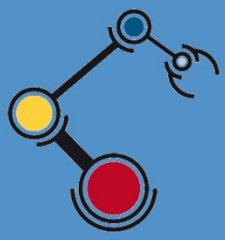
Escuela Politécnica Superior

# Diseño, simulación y control de un robot humanoide espacial



Máster Universitario en Automática y Robótica

Trabajo Fin de Máster

Autor:
Ignacio de Loyola Páez Ubieta

Tutor/es:
Jorge Pomares Baeza y Leonard Felicetti

Universitat d'Alacant
Universidad de Alicante

Julio 2021

# Diseño, simulación y control de un robot humanoide espacial


**Autor**

Ignacio de Loyola Páez Ubieta

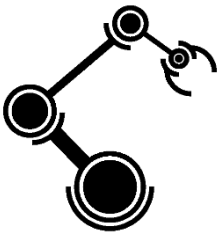
**Tutores**

Doctor Jorge Pomares Baeza

*Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal – Universidad de Alicante*

Doctor Leonard Felicetti

*Centro de sistemas autónomos y ciberfísicos – Cranfield University*

*Máster Universitario en Automática y Robótica*

ESCUELA POLITÉCNICA SUPERIOR

Universitat d'Alacant
Universidad de Alicante

ITER · FACITE · EIVS · QUAE · ASCENDIT · SVPER · OCCASVM
VNIVERSITAS LVCENTINA

*Alicante, Julio 2021*

# Resumen

Nuevos conocimientos, riquezas, trascendencia, vida más allá de los límites conocidos… Estos principios podrían identificarse como los objetivos de la ciencia o del propio ser humano, pero también fueron los principios que se tuvieron en el siglo XV cuando se decidió explorar el planeta o que se tienen para explorar el universo actualmente. Para explorar el universo se han empleado robots, los cuales tienen un papel importante debido a las condiciones extremas presentes en planetas extraterrestres. Mientras que los humanos han logrado poner un pie en la luna, se tienen conocimientos de las condiciones de planetas como Marte, Venus o Júpiter, los cuales ayudarán en futuras misiones espaciales. Hay muchos tipos de robots en el espacio, pero aquellos con forma humana son los que nos interesarán. Algunas tareas requieren que el robot se mueva de forma similar a los humanos, use las mismas herramientas que en misiones humanas o robóticas anteriores o la interacción con ellos sea más natural.

En primer lugar, se ha analizado el estado del arte en el que se encuentra actualmente la robótica o robots espaciales. Tras conocer sus orígenes, se ha pasado a clasificarlos en robots espaciales en órbita y robots planetarios. Tras esto, se ha profundizado en la clase de robot que se utilizará, la cual es un humanoide. Se tomará como robot de partida al robot TALOS de la compañía española PAL robotics.

Tras ello, entramos en la metodología y desarrollo del proyecto, en la cual se explicarán los pasos que se seguirán para poder tener un robot humanoide nuevo, totalmente funcional, para realizar tareas en gravedad 0. Para ello se contará con numerosas herramientas software y, gracias a ellas, con numerosos controladores diferentes.

Para comprobar si todos estos desarrollos matemáticos sirven de algo, se han realizado a modo de prueba 3 simulaciones con el simulador y se analizarán las gráficas que se extraen de las maniobras. Estas gráficas ayudarán a comprender si los cálculos realizados han llegado a buen cauce.

Por último, se aportarán las conclusiones que se han obtenido tras la realización del proyecto y se propondrán ampliaciones de este por si algún estudiante / docente se animara a continuar estos desarrollos o les sirvieran para avanzar en los suyos propios.

# Preámbulo: motivación, justificación y objetivo general

Si se trabaja duro, es posible alcanzar el techo. Una vez alcanzado, logra el cielo. Tras el cielo, ¿qué te impide no ir a por las estrellas? La vida es una lucha continua contra uno mismo por mejorar lo logrado hasta ese momento. Aunque uno se canse debe continuar, siempre hacia delante, en línea recta o esquivando los obstáculos que ponga el destino, ya que tras la muerte nos espera el descanso eterno.

Así pues, si un robot puede ayudarnos a preparar el cielo para cuando lleguemos al mismo, no se puede prescindir de dicha ayuda si el objetivo está más allá.

Tras un largo verano y el gran éxito del proyecto "Guiado de satélites robóticos mediante control basado en imagen" realizado como TFG del grado en ingeniería robótica en 2020, se decidió continuar con un proyecto en el mismo campo. Así pues, tras una breve reunión con Jorge Pomares en septiembre del 2020, se decidió en qué proyecto se trabajaría en los próximos meses.

Siempre se ha dicho que las segundas partes nunca fueron buenas, pero este TFM tiene el objetivo de ser la excepción que confirme dicha regla.

La decisión de realizar este máster fue precipitada por la mala situación sanitaria del país, además de la invitación por parte de Gabriel Jesús García, director del mismo, a que realizara la preinscripción. También fue influenciada por Andrés Úbeda, con quien se habló en febrero del 2020 y se llegó a la conclusión que este máster era uno de los mejores en su campo. Así pues, a finales de mayo del 2020, se supo que se realizaría.

Tras coger un robot humanoide terrestre, se ajustó para poder emplearlo en el espacio. Tras ello, se realizaron los experimentos en un simulador, ya que no se cuenta con suficiente material para poder montar este nuevo robot en la universidad, además de no tener una cámara de vacío en la que poner a flotar al robot para poder realizar los experimentos. No obstante, hoy en día los simuladores permiten obtener resultados altamente fiables, por lo que este impedimento de materiales no ha sido un problema para realizar el proyecto.

# Agradecimientos

En primer lugar, quería agradecer por segundo año consecutivo a Jorge Pomares y Leonard Felicetti por tutorizarme el proyecto final de estudios. Gracias Jorge por, a pesar de llevar este año tantos trabajos finales diferentes, hacerme un hueco y lograr que este proyecto haya salido adelante. Creo que una de las cosas que ha podido ser útil para todos ha sido aprender a cómo mejorar las máquinas virtuales para poder sacarles más rendimiento. Esto marcó un antes y un después en el proyecto, disminuyendo enormemente los tiempos de compilación y ejecución.

También me gustaría agradecer al equipo del máster haber logrado con los años un máster tan compacto, pero a la vez tan completo. Sinceramente, tenía mis dudas de que diese tiempo a todo, pero he podido comprobar de primera mano que si es posible. También es de agradecer la excelente sintonía con todos ellos, además de la gran disponibilidad para resolver dudas y satisfacer la curiosidad del alumnado.

No puedo olvidarme tampoco de Santiago Puente ya que, gracias al cual se consiguió ser portada de la revista del COITIA con el TFG presentado el año anterior.

Por último, quería agradecer a Carlos Villagra por seguir ahí un año más para lo que necesitase.

Pasando a la familia, debo agradecer a mis padres y hermana por aguantarme un año más en casa.

Ama, gracias por estar siempre cerca. Ya sabes que cuando necesites ayuda con lo que sea, solo tienes que pedirla (sobre todo con la tecnología). Espero heredar tu capacidad de trabajo algún día.

Papá, gracias por transmitirme que con paciencia todo se resuelve mejor. Creo que eres la persona más estudiosa que conozco, en concreto con las integrales. Deberás ayudar a María el año que viene.

María, gracias por ser tan distinta a mí. Estudia un poco más y tendrás el mundo a tus pies. Todos los genes que no me quedé yo, te los has quedado tu (úsalos con responsabilidad). Recuerda que ya hay mucha gente normal, aunque no tanta extraordinaria.

También agradecer a mis abuelos paternos por estar siempre presentes. Este último año ha sido difícil para todos nosotros. Sin embargo, más difícil que para ti, abuelo, lo dudo mucho. No todo el mundo tiene que volver a aprender a caminar. Cuando te sientas solo, piensa las noches de hospital que pasamos juntos. Tampoco me olvido de ti abuela. Aunque leas esto y a los 5 minutos ya no lo recuerdes, esto tiene su parte buena, siendo esta que te emociones cada vez que lo leas. Eso sí, lo malo de ello es que ya no recuerdes mi nombre.

Aitite, un año más te recuerdo. Se que te hubiese gustado haber visto que es de mi vida y que te sentirías muy orgulloso de cómo me va y de la persona en la que me he convertido. Se que, hace 6 años ya, te hice una promesa. No se cuando podré cumplirla, pero se que ya queda menos para ello. Recuerda que, aun así, si te presentas en algún momento, te seguiré acercando todo lo que me pidas y te llevaré a pasear tanto a pie como en coche. Y recuerda, somos los mejores, pase lo que pase.

José Santano, otro año más apareces aquí. No he conocido mucha gente de momento, pero creo que vas a ser una de las mejores personas que conoceré en mi vida. Gracias por haber coincidido conmigo.

A todos vosotros y a todos los que no he nombrado, gracias por haberme convertido en lo que soy y cauzar en lo que me convertiré. Parte de mi éxito será vuestro también.

*A ti, aitite, por no bajar nunca los brazos.*

*A mis padres, por enseñarme todo lo que soy.*

*Para aquellos que creen, ninguna prueba es necesaria.*

*Para aquellos que no creen, ninguna cantidad de pruebas es suficiente.*

San Ignacio de Loyola.

# Índice general

# Índice de figuras

# Índice de tablas

# 1. Introducción

Nuevos conocimientos, riquezas, trascendencia, vida más allá de los límites conocidos… Estos principios podrían identificarse como los objetivos de la ciencia o del propio ser humano, pero también fueron los principios que se tuvieron en el siglo XV cuando se decidió explorar el planeta (redescubriendo América o descubriéndola por parte de los europeos) o el universo.

Para explorar el universo se han empleado robots. Estos han tenido un papel importante debido a las condiciones extremas presentes en planetas extraterrestres. Una de las definiciones que se ha dado a los robots ha sido la de un sistema autónomo consistente en unidades electrónicas, eléctricas o mecánicas que puede sustituir a un ser vivo. Antes de los robots, los científicos enviaban animales, como perros o monos, para realizar experimentos, con el objetivo de incrementar el conocimiento que se tenía de otros planetas o satélites. Gracias a la mejora de la tecnología, hoy en día es posible realizar muchos de estos experimentos con robots, en detrimento de los seres vivos, preservando la vida. Mientras que los humanos han logrado poner un pie en la luna, se tienen conocimientos de las condiciones de planetas como Marte, Venus o Júpiter. A medida que avance la tecnología, se podrán obtener muchos más datos acerca de las condiciones del espacio. [1]

Hay muchos tipos de robots en el espacio, pero aquellos con forma humana son los que nos interesarán. Algunas tareas requieren que el robot se mueva de forma similar a los humanos (cambio de un filtro de aire), además de poder utilizar las mismas herramientas que en misiones humanas o robóticas anteriores (eliminando la necesidad de conectores o de la creación de herramientas para un único robot). [2]

## 1.1. Objetivos

Los objetivos del proyecto son:

- Diseño del robot: se estudiarán las características cinemáticas y dinámicas de este tipo de robots. El robot habrá de disponer de la suficiente capacidad de movimiento y manipulación para su aplicación a este tipo de tareas. Dispondrá de dos brazos con manos robóticas, así como de un torso para asimilar su espacio de trabajo, similar al de un humano.

- Simulación del robot: una vez diseñado el robot se habrán de simular las condiciones orbitales de ausencia de gravedad / gradiente orbital para asemejar su comportamiento al que dispondría en órbita. Además, se empleará gazebo para simular el robot y un entorno de funcionamiento lo más realista posible.

- Control del robot: se evaluará el robot realizando distinto tipo de tareas (alcance, manipulación, uso de herramientas, etc.). Para ello será necesario no sólo definir las posiciones a alcanzar sino también las trayectorias necesarias. Se utilizará ROS Control para evaluar el comportamiento del robot utilizando controladores PID articulares clásicos, así como otras estrategias de control como control por Par calculado, PD+, etc. Esto permitirá evaluar y determinar el controlador más adecuado para este tipo de aplicaciones.

## 1.2. Estructura de la memoria

La memoria del TFM se divide en seis partes diferentes.

Comenzamos con una introducción, en la cual se analizan las razones para explorar el espacio, por qué hacerlo con robots humanoides, los objetivos que persigue el proyecto y la estructura de la memoria.

En la segunda parte de la memoria se explicará que es un robot, una breve historia de los robots espaciales y que son los robots humanoides. Tras ello, se analizará la empresa a la que pertenece el robot original y el robot que se ha diseñado a partir de este para el trabajo. Para concluir esta parte, se analizará que software ha sido necesario emplear para poder llevar a cabo el TFM.

A continuación, en la tercera parte de la memoria, se verá el procedimiento seguido para desarrollar el TFM. Se verá que prácticamente todas las herramientas usadas son software, además del lenguaje de programación que se empleó para desarrollarlo.

Seguido, en la cuarta parte de la memoria, se desarrollará el entorno de simulación que se empleará, los ajustes pertinentes al robot para emplearlo en el espacio y los controladores que se usarán para realizar los movimientos.

En la quinta parte de desarrollan 3 experimentos diferentes para probar si el robot ha sido diseñado de forma correcta. Para ello, se emplearán controladores articulares y cartesianos.

En la sexta parte se presentarán las conclusiones del proyecto, además de posibles trabajaos futuros que podrían realizarse a partir de este proyecto.

# 2. Marco teórico

En esta segunda parte veremos algunos conceptos básicos, como son la definición de robots, cuándo surge el término que se emplea hoy en día, una breve historia de los robots espaciales y como se pueden clasificar, y que es un robot humanoide y cuáles han viajado o se encuentran actualmente en desarrollo para viajar al espacio.

Tras ello, se hablará de la compañía española PAL Robotics. Esta empresa ha diseñado y diseña varios robots humanoides. Uno de ellos, en concreto el robot TALOS, será en el que se profundice un poco más, ya que se empleará en este proyecto. Se analizarán sus características técnicas y se verá como de extendido esta su uso por el mundo.

Por último, se verá que software se ha empleado. Este incluye ROS, gazebo, URDF y XACRO, ROS Control, KDL, Rosbag y Plotjuggler.

## 2.1. Robot

Un robot es un sistema mecánico multipropósito, reprogramable y controlado automáticamente con múltiples grados de libertad, pudiendo ser fijos o móviles [3]. Como toda tecnología sofisticada, la robótica es el resultado de varios cientos de años de conocimientos acumulados, relativos a leyes físicas, matemáticas y geométricas.

El origen de la palabra tiene su nacimiento en 1921. En este año, el escritor checo Karel Čapek presentó su obra Rossum's Universal Robots (R.U.R.) [4]. El cartel de dicha obra puede verse en la figura 2.1. En ella aparecían trabajadores humanoides, los cuales cumplían la misma función que los robots que, posteriormente, se incluyeron en las fábricas unas décadas más tarde (liberando a los humanos de trabajos difíciles, monótonos o peligrosos).

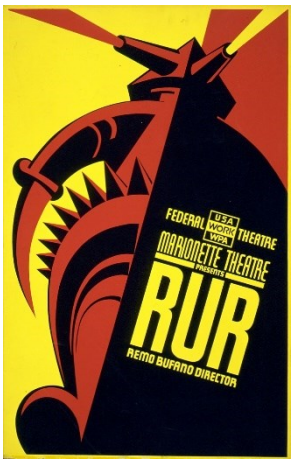



**Figura 2.1:** Cartel de la obra Rossum's Universal Robots de 1921. Fuente: Wikimedia commons

Respecto a sus campos de aplicación, son diversos y diferentes, contando entre ellos con la automatización industrial, aplicaciones médicas, transporte, entretenimiento, aéreas, submarinas o espaciales.

## 2.2. Robótica espacial

Los robots espaciales son aquellos robots que operan en el medio ambiente espacial. Desde los inicios de la carrera espacial y de la exploración para la conquista del espacio, a principios de la década de 1960, se ha hecho uso de robots como apoyo a ciertas actividades espaciales, los cuales, si bien primitivos en sus inicios, han aumentado grandemente su capacidad y calidad de operación, mejorando su diseño, desempeño, características y complejidad tecnológica para su uso en el espacio.

Estos poseen mayor fuerza, velocidad, precisión, seguridad y economía de uso con respecto a los humanos, así como la falta de necesidad de descanso, vestido, alimentación o atmósfera. Esta ventaja hace que los robots espaciales sean sumamente atractivos para su operación y apoyo fuera de la Tierra. Gracias a ellos, la dificultad, complejidad, costo y peligro se reducen grandemente al sustituir a los humanos en el espacio exterior. Los robots pueden ser diseñados para soportar la temperatura extrema, radiación, microgravedad y vacío presentes en el espacio exterior, reduciendo los riesgos para los astronautas fuera de la Tierra [5].

Se pueden dividir en dos ramas diferentes, de las cuales puede verse un ejemplo en la figura 2.2: robots espaciales en órbita y robots planetarios.

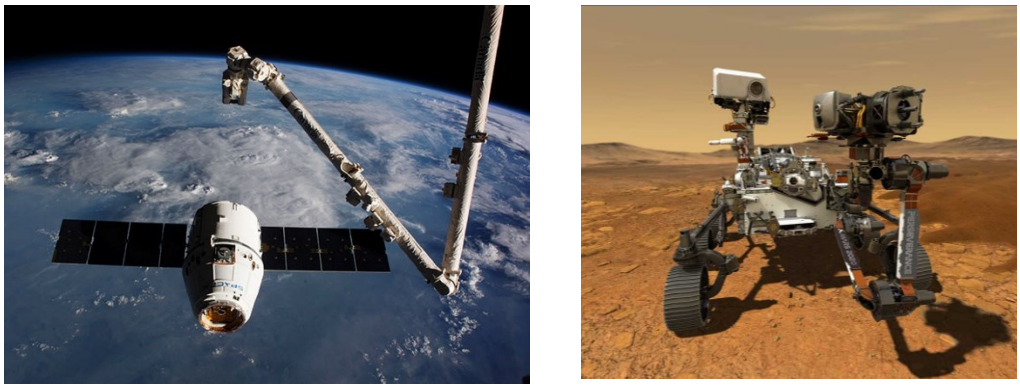


**Figura 2.2:** Robot espacial en órbita Canadarm 2 capturando la nave de carga Dragon de Space X (izquierda) y robot planetario Perseverance, rover recién aterrizado en Marte (derecha). Fuentes: http://spaceref.com/ y https://mars.nasa.gov/.

## 2.3. Robots humanoides en el espacio

Dentro del grupo de robots espaciales en órbita, aparecen los robots humanoides. Estos pueden ser definidos como máquinas programables, las cuales pueden imitar las tareas realizadas por los humanos, además de su apariencia. La investigación en torno a estos robots ayuda a los investigadores a entender más acerca del comportamiento y estructura humana. Tras muchos años de investigación y desarrollo, se pueden encontrar en diferentes tamaños, formas y capacidades. La mayoría de humanoides poseen un torso, dos piernas, dos manos y una cabeza. No obstante, en ciertas aplicaciones, es posible prescindir de alguno de los elementos nombrados anteriormente [6].

A continuación, se verán algunos de los ejemplos de robots humanoides que se encuentran en el espacio [7]:

- Robonaut: diseñado por la NASA, en colaboración con DARPA. Su primera versión no salió de la Tierra, aunque sirvió para probar la tecnología y aplicarlo a su segunda versión. Robonaut II ha estado en la ISS desde 2011, sustituyendo a los astronautas en la realización de tareas tediosas, contando con unas piernas a modo de manipuladores especiales de escalada para sujetarse a las superficies. La siguiente versión del Robonaut podrá ser usado en paseos espaciales. Este robot puede verse en la figura 2.3.

**Figura 2.3:** Robonaut II, creado por la NASA y DARPA. Fuente: https://es.wikipedia.org/.

- Valkyrie / R5: diseñado en una competición patrocinada por la DARPA en 2011 para realizar tareas de asistencia en caso de desastre, así como operaciones de búsqueda y rescate. En 2015 fue estudiado por la NASA, junto a estudiantes del MIT y de la Northeastern University, para ver cómo se podría emplear en tareas de exploración espacial. Actualmente se encuentra en la Tierra, pero ha demostrado una flexibilidad y destreza increíbles, por lo que tal vez pueda verse en Marte algún día. Este robot puede verse en la figura 2.4.

**Figura 2.4:** Valkyrie, creado por la DARPA. Fuente: https://www.nasa.gov/.

- RoboSimian: similar a un simio. Puede mapear el entorno en 3D gracias a la tecnología LIDAR. Es extremadamente flexible y puede ir por terrenos arduos o realizar tareas que requieran gran destreza. Según la NASA, podría ser usado en tareas de recuperación tras un desastre. Participó en una competición patrocinada por la DARPA en 2015. Este robot puede verse en la figura 2.5.

**Figura 2.5:** RoboSimian, creado por la DARPA. Fuente: https://www.space.com/.

- Kirobo: la JAXA envió un mini astronauta de 34 cm de altura para probar la interacción humano-robot. Fue a la ISS en 2013, llegando a mantener una conversación en japonés con el astronauta Koichi Wakata. Fue construido en los laboratorios de investigación de la Universidad de Tokio de ciencias y tecnologías avanzadas, en colaboración con Toyota. También puede reconocer voces, caras y emociones. Permitirá descubrir nuevas maneras de combatir la soledad orbital. Este robot puede verse en la figura 2.6.

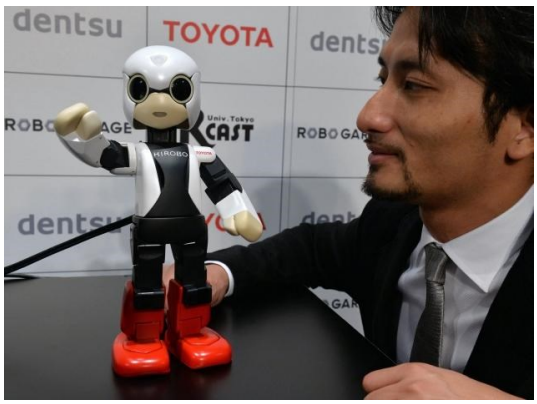



**Figura 2.6:** Kirobo, creado por la Universidad de Tokio y Toyota. Fuente: https://www.puroperiodismo.com/.

- AILA: diseñado en el centro alemán de investigación para la inteligencia artificial, tiene como objetivo mejorar el trabajo robótico tanto dentro como fuera de la ISS, en la Luna o Marte. Para ello imita las emociones humanas, analiza los movimientos humanos, los divide en cortos segmentos y los almacena en una librería para después generar otros nuevos. Este robot puede verse en la figura 2.7.

**Figura 2.7:** AILA, creado por centro alemán de investigación para la inteligencia artificial. Fuente: https://robots.ieee.org/.

- Vyommitra: se trata de un robot semi-humanoide que la ISRO planea enviar a la Luna a bordo de una misión no tripulada. Será capaz de dar avisos si el ambiente de la cabina cambia, simular algunas posturas humanas, operar los paneles de mando o reconocer y charlar con sus compañeros humanos (en el futuro). Este robot puede verse en la figura 2.8 [8].

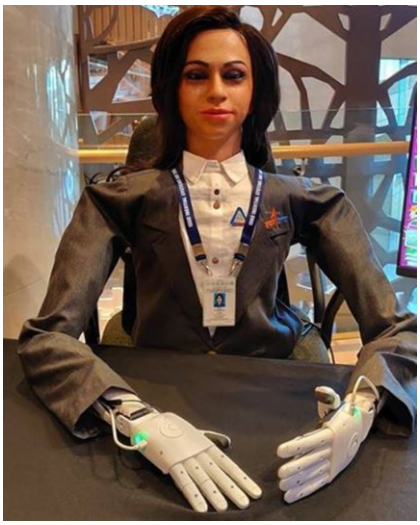


**Figura 2.8:** Vyommitra, creado por la ISRO. Fuente: https://indianexpress.com/.

- Fedor: fue diseñado para realizar tareas de exploración y rescate, además de viajes espaciales. Fue presentado en 2017, midiendo 1.8 m y pesando 160 kg. Se trata de un robot totalmente autónomo, aunque es posible teleoperarlo. Fue enviado por la Roscomos a la ISS en 2019 y, bajo la supervisión del astronauta ruso Alexander Skvortsov, realizó diversas tareas. No viajará más al espacio, ya que se vio que era demasiado corpulento y tenía una baja capacidad de manipulación. Este robot puede verse en la figura 2.9 [9].

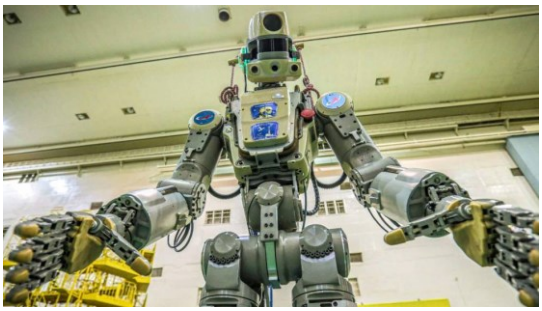


**Figura 2.9:** Fedor, creado por Android Technics. Fuente: https://elpais.com/.

## 2.4. PAL Robotics

PAL Robotics es una empresa originaria de Barcelona (España). Se encuentra en el corazón del distrito tecnológico de Barcelona, a corta distancia del conocido paseo de Las Ramblas. Se encarga de diseñar, programar y montar robots de servicio que pueden, entre otras tareas, ayudar a aumentar la productividad en el trabajo, llevar las maletas al aeropuerto o ayudar a personas de la tercera edad. El logo de esta empresa puede verse en la figura 2.10 [10].

Comenzó en 2004, cuando un pequeño grupo de ingenieros construyó el primer robot humanoide bípedo completamente autónomo de Europa. Con más de 15 años de experiencia en I+D, hoy son conocidos por los robots humanoides y móviles que se usan en investigación, tareas de intralogística, automatización de inventario y aplicaciones sociales para empresas, instituciones y laboratorios de I+D de todo el mundo [11].

A modo de cronología, se han desarrollado varios robots desde su nacimiento:

- REEM-A (2005).
- REEM-B (2008).
- REEM-H1 (2010).
- REEM (2012).
- REEM-C (2013).
- StockBot & TIAGo (2015).
- TALOS (2017).
- ARI (2019).


**Figura 2.10:** Logo de la empresa PAL Robotics. Fuente: https://pal-robotics.com/.

## 2.5. El robot TALOS

Para este TFM se ha escogido como base el robot TALOS. Este robot fue creado por la empresa PAL Robotics en el año 2017, el cual puede verse en la figura 2.11.

**Figura 2.11:** Robot TALOS de PAL Robotics. Fuente: https://pal-robotics.com/.

### 2.5.1. Especificaciones generales

El robot TALOS cuenta con las siguientes especificaciones técnicas [12]:

- Mide 175 cm de altura y pesa 95 kg.
- Cuenta con 32 GDL repartidos de la siguiente manera: 6 en cada pierna, 7 en cada brazo, 1 en la pinza, 2 en el cuello y 2 en la cadera.
- Puede realizar manipulación de objetos de hasta 6 kg.
- Cuenta con conectividad Wi-Fi 802.11 a/b/g/n, Ethernet y EtherCat.
- Tiene una autonomía de 1.5 horas en movimiento y 3 horas en modo de espera. Para ello tiene una batería de iones de litio de 1080 Wh, con una descarga superior a los 100 A.
- Para comunicarse con los humanos, ya que cuenta con altavoces y 24 LEDs RGB.
- El robot tiene sensores de fuerza y par, además de IMUs.
- Respecto a su capacidad de cómputo, cuenta con 2 intel core i7. Su sistema operativo es Ubuntu LTS.

### 2.5.2. Extensión de uso

Este robot en particular supone un coste aproximado de 900000 €. No obstante, es posible encontrarlo en varios centros de investigación como en el LAAS-CNRS (Toulouse, Francia), la Universidad de Edimburgo (Escocia, Reino Unido) o la Universidad de Waterloo (Ontario, Canadá). En cada ubicación el robot ha sido "personalizado", tal y como puede verse en la figura 2.12 [13] [14] [15] [16].

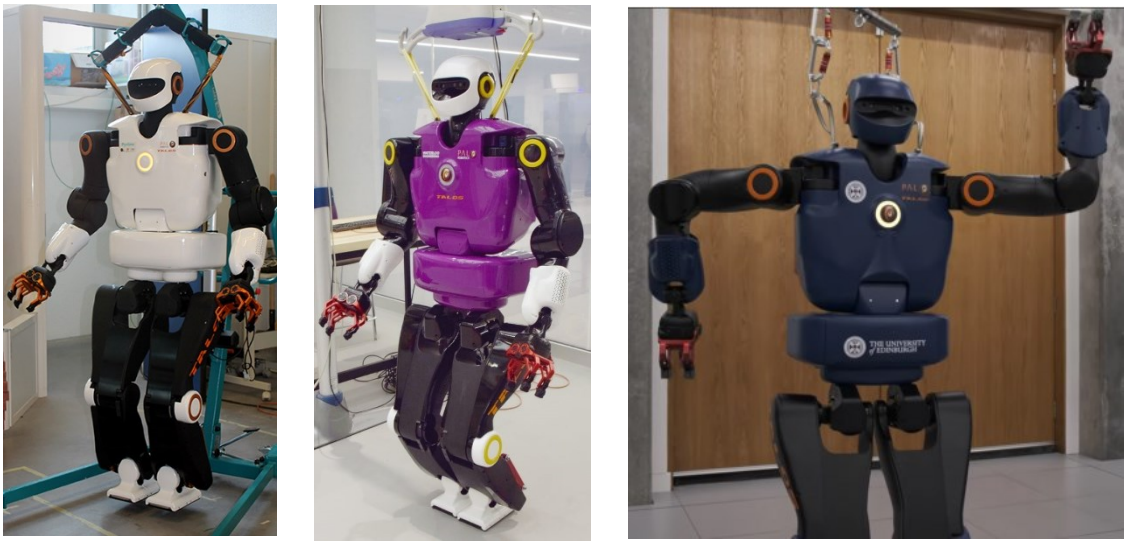


**Figura 2.12:** Robot TALOS en el LAAS-CNRS, en la Universidad de Edimburgo y en la Universidad de Waterloo (de izq a der). Fuentes: https://www.ifac2017.org/, https://www.cbc.ca/ y https://www.ed.ac.uk/.

## 2.6. Software

En este apartado se verán que herramientas de software se han empleado para poder desarrollar el proyecto. Estas incluyen ROS, Gazebo, URDF y XACRO, ROS control, KDL, rosbag y plotjuggler.

### 2.6.1. ROS

ROS es un framework flexible para escribir software robótico. Fue creado porque realizar software robótico es complejo, siéndolo tanto que no hay persona, laboratorio o institución capaz de realizarlo por sí mismo. Como resultado, ROS fue construido desde cero, animando al desarrollo de software robótico de forma colaborativa. Así pues, la colaboración de varios grupos expertos en temas totalmente diferentes puede crear software muchos más avanzado, partiendo del trabajo de otros expertos. Su logo puede verse en la figura 2.13 [17].

Su origen se remonta al año 2000, cuando la Universidad de Standford consiguió crear los programas STAIR y PR. En 2007, Willow Garage proporcionó muchos recursos y creó implementaciones bien probadas. Este esfuerzo fue impulsado por numerosos investigadores, los cuales usaron su tiempo y experiencia en la creación del núcleo de ROS y los paquetes de software fundamentales. El software era desarrollado abiertamente, usando licencias open source.

Hoy en día, el ecosistema de ROS consiste en miles y miles de usuarios alrededor del mundo, trabajando en proyectos que van desde hobbies hasta amplios sistemas de automatización industriales [18].


**Figura 2.13:** Logotipo de ROS. Fuente: https://www.ros.org/.

### 2.6.2. Gazebo

Gazebo, que nace con el nombre de Gazebo Project, es un simulador 3D, cinemático, dinámico y multi-robot que permite realizar simulaciones de robots articulados en entornos complejos, interiores o exteriores, realistas y tridimensionales.

Entre sus características podemos nombrar las siguientes:

- Gazebo es un software libre financiado por Willow Garage.
- Podemos ejecutar Gazebo desde ROS y utilizar las APIs de este último para controlar los robots en las simulaciones.
- Simulación realista de la física de los cuerpos rígidos
- Capacidad de desarrollar y simular modelos de robots propios (URDF).
- Posibilidad de crear escenarios (mundos) de simulación.
- Contiene diversos plugins para añadir sensores al modelo del robot y simularlos.

El logo de la empresa y un ejemplo de entorno pueden verse en la figura 2.14 [19].

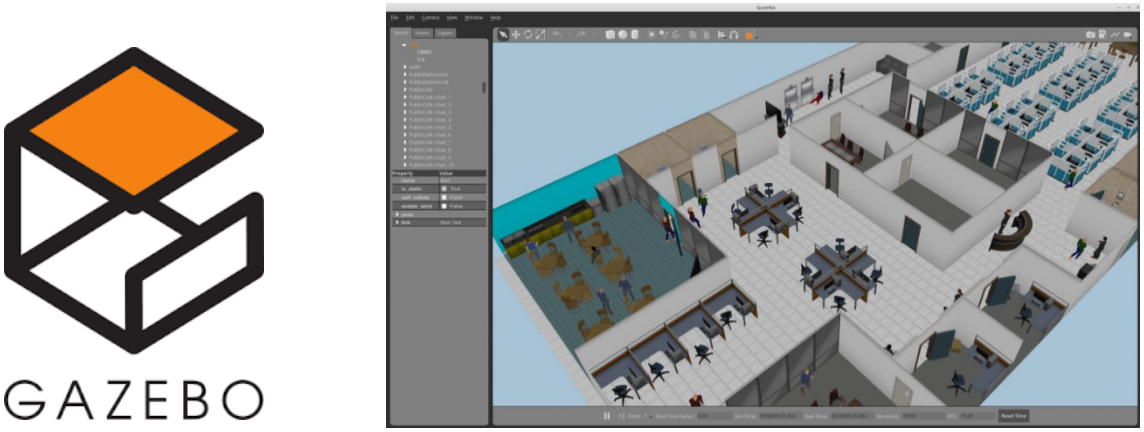



**Figura 2.14:** Logotipo de Gazebo y ejemplo de ambiente creado. Fuentes: http://gazebosim.org/ y https://medium.com/.

### 2.6.3. URDF y XACRO

URDF es un formato especificado en XML compatible con diversos simuladores, entre ellos con Gazebo, siendo además el formato empleado en ROS para el modelado de robots. Consta de una serie de etiquetas XML que permiten especificar una serie de parámetros de cada eslabón y articulación del robot, formando un árbol de eslabones (links) conectados entre sí mediante articulaciones (joints) que determinan el parentesco entre ellos. Un ejemplo de árbol puede verse en la figura 2.15. Estas articulaciones pueden ser fijas o móviles, siendo las móviles a su vez rotacionales, lineales o flotantes. Actualmente, únicamente es posible emplearlo en robots de tipo serial (dejando fuera las cadenas cinemáticas cerradas). Este formato permite especificar la descripción cinemática y dinámica de los eslabones y articulaciones del robot, la representación visual de los eslabones y el modelo de colisión de cada eslabón.

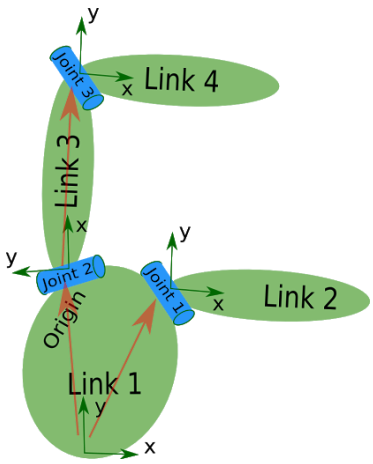



**Figura 2.15:** Ejemplo de árbol de eslabones conectados mediante articulaciones. Fuente: http://library.isr.ist.utl.pt/.

Como su nombre indica, XACRO es un lenguaje macro. Este programa ejecuta todas las macros y genera un resultado, pudiendo construir archivos XML más cortos y legibles utilizando macros que se pueden expandir a expresiones XML más grandes. Mejora a URDF, ya que permite incluir constantes, operaciones matemáticas simples y crear macros. El objetivo principal de su uso es el de reducir el tamaño general del archivo URDF y facilitar su lectura y mantenimiento. Así pues, el programa de lectura de archivos XACRO lee el archivo, ejecuta todas sus macros y genera el resultado en archivo URDF final [20].

### 2.6.4. ROS control

ROS Control es un conjunto de paquetes y herramientas que permiten enviar comandos y comunicarse con las articulaciones y actuadores de un robot, con el objetivo de controlarlos. Para ello, son necesarios los interfaces de los controladores, los manejadores de los controladores, transmisiones, interfaces hardware y la toolbox de control. Un esquema de ROS Control puede verse en la figura 2.16.

Se emplea como entrada los datos de estado de las articulaciones y una referencia articular. Además, estos paquetes generan como salida el comando adecuado para los actuadores, con el objetivo de que los valores articulares actuales (joint states) alcancen la referencia articular.

Los paquetes ros_control proporcionan varios controladores (controller plugins). Estos tienen un comando de entrada (referencia) y uno de salida (acción de control). Los comandos de salida están relacionados con la interfaz articular (joint interface), que depende del tipo de articulación que se esté controlando (se habrá de seleccionar uno u otro, excepto en el caso de joint_state_controller, que siempre se inicia). Se puede también cargar más de un plugin (esfuerzo, posición y/o velocidad) por robot, y cambiar entre ellos usando el controller_manager.

Una vez que los plugins han hecho su labor, la salida de estos es enviada al interfaz hardware (que actúa como un intermediario entre los plugins y el robot real o simulado) [21].

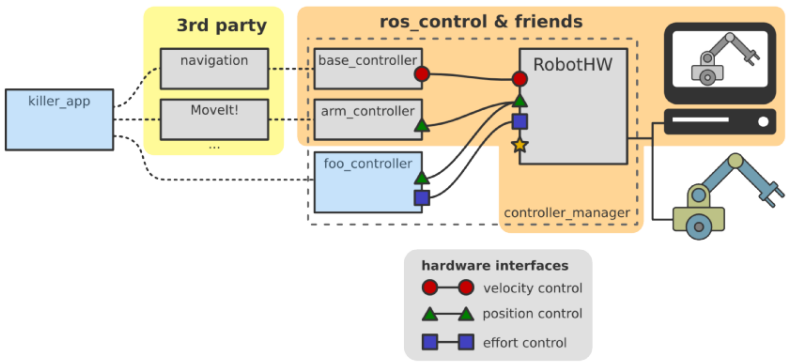



**Figura 2.16:** Esquema de ROS control. Fuente: https://roscon.ros.org/.

### 2.6.5. KDL

KDL es una librería distribuida por el proyecto Orocos que puede usarse con ROS. Es una aplicación para modelar y realizar cómputos de cadenas cinemáticas. KDL incluye además una serie de funciones que permiten trabajar con vectores, puntos, transformaciones de sistemas de coordenadas, etc. Contiene definidas una serie de clases y funciones que permiten resolver la cinemática directa e inversa del robot, su Jacobiano, las diferentes componentes del modelo dinámico (vectores de gravedad o inercia) o, junto a URDF, extraer su modelo cinemático y dinámico. Un ejemplo de esto último puede verse en la figura 2.17.

El stack de KDL está dividido en 3 paquetes, siendo estos KDL (contiene la versión más reciente de la librería KDL), Orocos_KDL (versión en C++ de la librería) y PyKDL (versión en Python de la librería) [21] [22].

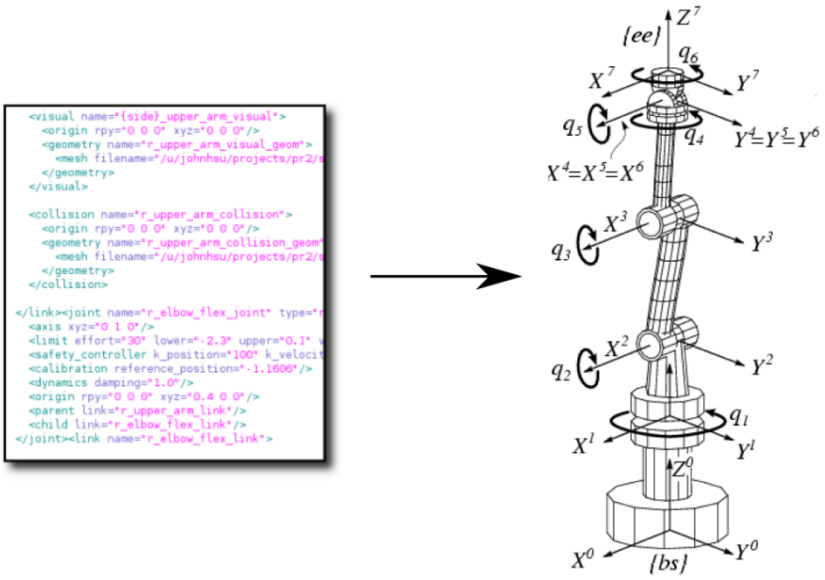



**Figura 2.17:** Obtención del modelo del robot a partir del código URDF. Fuente: http://library.isr.ist.utl.pt/.

### 2.6.6. Rosbag

Rosbag es una serie de herramientas que permiten guardar la información de los topics de un robot para poder reproducirlas en otro momento. Se puede ocupar, entre otras cosas, para guardar movimientos y acciones del robot para poder usarla en experimentos posteriores.

El comando rosbag tiene una serie de instrucciones, entre las cuales destacan [23] [24]:

- Record: este comando permite guardar topics, los cuales se les pasan en orden. Se puede usar la opción -a para guardarlos todos.
- Play: permite reproducir un rosbag previamente guardado con record. Cuando se reproduce el rosbag, se puede pausar usando la tecla espacio, luego se puede avanzar paso por paso con la tecla s.
- Compress: debido a la cantidad de información, este comando permite comprimir los archivos utilizados.
- Decompress: descomprime los archivos comprimidos previamente con el comando compress.
- Info: muestra un resumen del contenido de un fichero bag.

### 2.6.7. PlotJuggler

Se trata de una herramienta para visualización de series temporales de datos de forma rápida, intuitiva y extensible.

Cuenta con una interfaz que funciona de modo “drag & drop”, diseñada para maximizar la rapidez y simplicidad de uso. Puede emplearse para visualizar logs (datos off-line y on-line) en varios campos, como son la robótica, la conducción autónoma, la ciencia de datos, la automatización o los sistemas embebidos.

Permite añadir nuevas funcionalidades y fuentes de datos gracias a su arquitectura basada en plugins. De esta manera, se puede añadir una capa de transporte propia y / o protocolos. Además, es compatible con ROS y ROS2, con los ficheros CSV, con el ploteo de datos en vivo usando MQTT, ZeroMQ o sockets y los protocolos que soporta son, entre otros, JSON, BSON, o CBOR.

Por si fuese poco, se pueden aplicar funciones y transformaciones a los datos para entenderlos mejor. Entre ellas se encuentran la derivada, la integral o las escalas. Se basa en el lenguaje LUA para la generación de nuevas funciones propias más complejas.

Por último, permite integrarlo con ROS. Así pues, se permite la carga de ficheros rosbag, suscripción a topics o la republicación de mensajes y su posterior visualización con RViz [25].


**Figura 2.18:** Logo del programa PlotJuggler. Fuente: https://index.ros.org/.

# 3. Metodología

En este apartado de la memoria, se verá que pasos se siguieron a la hora de desarrollar este TFM. A pesar de comenzar antes a trabajar en él, siempre se pueden realizar más y más cosas. Así pues, se describirán de forma aproximada que pasos se realizaron.

Tras ello, se verán las herramientas que se han empleado. Prácticamente todas las herramientas son software, ya que es complicado disponer de robots tan avanzados en las universidades hoy en día, además de cámaras de vacío para realizar simulaciones reales. Además, se explicarán las razones por las que se usó como lenguaje de programación C++.

## 3.1. Pasos del desarrollo

Tras la reunión con Jorge Pomares en septiembre del año pasado, se comenzó a desarrollar este TFM. No obstante, a pesar de empezar 5 meses antes a trabajar en él, parece que siempre falta tiempo (nunca se acaba).

Se comenzó viendo algunos tutoriales en la web "the construct". Esto se hizo para ver cómo manejar en ROS los robots de la empresa PAL Robotics. Una vez entendido, se procedió a instalar todos los paquetes necesarios en una máquina virtual para poder lanzar el robot TALOS en Gazebo. Tras lograr ver el robot TALOS en Gazebo, se procedió a ajustarlo para poder enviarlo al espacio. Esto consistió en modificar la gravedad del simulador, eliminar las piernas y pies del robot y cambiar las pinzas de dicho robot por unas manos robóticas, desarrolladas en el modelo REEM-C. Una vez se tiene el robot que se empleará en el trabajo, fue el momento de probar a moverlo. Tras muchos intentos fallidos, se logró crear controladores propios y aplicarlos al robot. Lo siguiente fue incorporar la ISS a la simulación de Gazebo. Se intentó coger alguna de las asas repartidas por ella con controladores de posición, pero no se logró. Para intentar mitigar ese error, se creó un controlador cartesiano con un punto inicial y final. También se intentó crear controladores de trayectorias, pero los paquetes originales no se consiguieron compilar, por lo que no se pudieron crear finalmente.

Todas estas tareas pueden resumirse en los siguientes pasos:

- Entender ROS y los paquetes de la empresa PAL Robotics.
- Ajuste del simulador y creación del robot.
- Creación de controladores de posición / fuerza.
- Introducir la ISS a la simulación y realizar tareas de grasping.
- Creación de controladores cartesianos y de trayectorias.

## 3.2. Herramientas usadas

Para poder realizar todas las tareas nombradas previamente, se empleó como hardware un ordenador portátil únicamente, ya que no se contaba en la universidad con un robot TALOS, las manos de un robot REEM-C, ni con una cámara de vacío para poder simular las condiciones espaciales. Aunque tuviesen en la universidad dichos robots, dudo mucho que me permitieran desmontarlos para crear el robot de esta simulación.

No obstante, las herramientas software son muy variadas e igualmente efectivas:

- Acceso a la web de cursos https://www.theconstructsim.com/.
- Ubuntu 16.04.4.
- ROS Kinetic.
- Gazebo.
- Paquetes de simulación del robot TALOS.
- Paquetes de simulación de las manos hey5 del robot REEM-C.
- URDF / XACRO.
- ROS Control.
- KDL.
- Paquetes de simulación de la ISS.
- Rosbag.
- Plotjuggler.

Respecto al lenguaje de programación para emplear estas herramientas, se eligió C++ y XML. Se ha elegido C++ como lenguaje de programación principal ya que [26]:

- El código se ejecuta muy rápido.
- Gracias a tener que compilar los códigos realizados, se pueden detectar muchos errores antes de lanzar el programa.
- C++ está muy extendido, pudiendo encontrar un número infinito de bibliotecas que te permiten realizar todo lo que desees.
- Es el lenguaje usado en la industria robótica.

# 4. Desarrollo

En este apartado de la memoria se verá el entorno de simulación desarrollado para la aplicación. Se verán las diferentes gravedades en el sistema solar y se elegirá la gravedad 0 como modelo (al producirse situaciones de ingravidez, ya que no existe ningún punto del universo no afectado por la gravedad).

Tras ello, se diseñará el robot que se empleará en los experimentos. Por esa razón, el robot TALOS debe ser ajustado para la tarea que realizará en gravedad 0. Estos ajustes consisten en eliminar las piernas, las pinzas, dotarle de manos y personalizar los colores de sus piezas. La última personalización es juntarlo con la ISS en el simulador.

Por último, se explicarán los diferentes controladores que se analizarán para realizar los movimientos. Estos pueden ser articulares (distinguiendo el P, PD, PD con compensación de gravedad, PD con compensación precalculada de gravedad y PID) o cartesianos (basado en velocidades).

## 4.1. Entorno de simulación

En el apartado anterior se explicó el simulador Gazebo. En dicho simulador se realizarán todas las pruebas pertinentes con el robot que posteriormente se explicará.

Se partió de los paquetes del robot TALOS de la empresa PAL Robotics. Estos paquetes están publicados en el repositorio de github de la empresa, la cual apuesta por proporcionar códigos para poder, a partir de ellos, crear simulaciones propias ajustadas a la aplicación que se investigue.

Una vez descargados, se procedió a buscar en que fichero se indicaba la gravedad que afecta al robot en la simulación. Una vez localizado, se comprobó que estaba fijada a 9.81 $^{m}/_{s^2}$ en la dirección del eje z, en sentido negativo. Esta es la gravedad media que se tiene en la Tierra.

```
<physics type="ode">
  <gravity>0 0 -9.81</gravity>
  <ode>
    <solver>
      <type>quick</type>
      <iters>50</iters>
      <sor>1.4</sor>
    </solver>
    <constraints>
      <cfm>0.0</cfm>
      <erp>0.2</erp>
      <contact_max_correcting_vel>100.0</contact_max_correcting_vel>
      <contact_surface_layer>0.0</contact_surface_layer>
    </constraints>
  </ode>
  <real_time_update_rate>1000</real_time_update_rate>
  <max_step_size>0.001</max_step_size>
</physics>
```

**Figura 4.1:** Gravedad por defecto de la simulación.

La fuerza de gravedad es posible calcularla usando la ley de la gravitación universal y la segunda ley del movimiento. Estas vienen dadas por (4.1) y (4.2).

$$F_{21} = -G \frac{m_1 m_2}{|r_2 - r_1|^2} \hat{u}_{21} \tag{4.1}$$

En ella pueden verse los términos:

- $F_{21}$: fuerza de la gravedad.
- $G$ : constante de gravitación universal de Newton, equivalente a $6.672 \times 10^{-11}\ N\,m^2/_{kg^2}$.
- $m_1,\ m_2$: masa de partículas puntuales 1 y 2.
- $\hat{u}_{21}$: vector unitario que va dirigido de la partícula 1 a la 2.
- $r_2 - r_1$: distancia que separa ambas partículas.

$$F = m\,a \tag{4.2}$$

En ella pueden verse los términos:

- F: fuerza.
- $m$: masa de la partícula.
- $a$: masa de la partícula.

La fórmula 4.2 es válida únicamente si la masa del cuerpo se mantiene constante.

Combinando ambas, nos queda una nueva fórmula, la cual nos permitirá calcular la aceleración de la gravedad. Esta puede verse en (4.3).

$$a = G\,\frac{M}{r^2} \tag{4.3}$$

Así pues, si se tienen la masa y el radio de un cuerpo, es posible calcular la aceleración de la gravedad a la que está sometido un objeto. En la tabla 4.1 puede verse algunas masas, radios y gravedades de varios cuerpos celestes interesantes [27].

| Lugar | Masa (kg) | Radio (m) | Aceleración de gravedad ($^{m}/_{s^2}$) |
|---|---|---|---|
| Mercurio | $3.30 \times 10^{23}$ | $2.44 \times 10^{6}$ | 3.70 |
| Venus | $4.87 \times 10^{24}$ | $6.05 \times 10^{6}$ | 8.90 |
| Tierra | $5.97 \times 10^{24}$ | $6{,}38 \times 10^{6}$ | 9.81 |
| ISS | $5.97 \times 10^{24}$ | $6.78 \times 10^{24}$ | 8.96 |
| Luna | $7.3 \times 10^{22}$ | $3.65 \times 10^{4}$ | 1.60 |
| Marte | $6.42 \times 10^{23}$ | $3.40 \times 10^{6}$ | 3.70 |
| Júpiter | $1.90 \times 10^{27}$ | $7.15 \times 10^{7}$ | 23.1 |
| Saturno | $5.68 \times 10^{26}$ | $6.03 \times 10^{7}$ | 9.0 |
| Urano | $8.68 \times 10^{25}$ | $2.56 \times 10^{7}$ | 8.7 |
| Neptuno | $1.02 \times 10^{26}$ | $2.48 \times 10^{7}$ | 11.0 |
| Plutón | $1.46 \times 10^{22}$ | $1.19 \times 10^{6}$ | 0.7 |
| Vacío | 0.00 | | 0.0 |

**Tabla 4.1:** Aceleración de la gravedad en varios lugares del sistema solar.

Tras ver la tabla 4.1, se ha visto conveniente realizar dos aclaraciones [28]:

- En la ISS hay una gravedad importante creada por la Tierra: los astronautas flotan debido a que se produce un estado de ingravidez, que es un estado en el que un cuerpo que tiene un cierto peso se contrarresta totalmente con otra fuerza o se mantiene en caída libre sin sentir los efectos de la fuerza gravitatoria.
- La gravedad en el vacío es existente, aunque sea un valor muy pequeño. A modo de ejemplo, el efecto de la gravedad de la Luna influencia las mareas terrestres.

Así pues, es posible realizar muchas simulaciones en varios entornos de gravedad. No obstante, al desear crear un robot que realizara tareas en el espacio, se asume una gravedad nula, a pesar de que no se dará nunca (pero será el peor de los casos que se darán). En la imagen 4.2 se puede ver esta nueva gravedad nula impuesta.

```
<physics type="ode">
  <gravity>0 0 0</gravity>
  <ode>
    <solver>
      <type>quick</type>
      <iters>50</iters>
      <sor>1.4</sor>
    </solver>
    <constraints>
      <cfm>0.0</cfm>
      <erp>0.2</erp>
      <contact_max_correcting_vel>100.0</contact_max_correcting_vel>
      <contact_surface_layer>0.0</contact_surface_layer>
    </constraints>
  </ode>
  <real_time_update_rate>1000</real_time_update_rate>
  <max_step_size>0.001</max_step_size>
</physics>
```

**Figura 4.2:** Nueva gravedad impuesta al simulador.

No obstante, tal y como puede verse en la figura 4.3, si en gravedad 0 no hay ninguna acción, no se produce ninguna reacción.

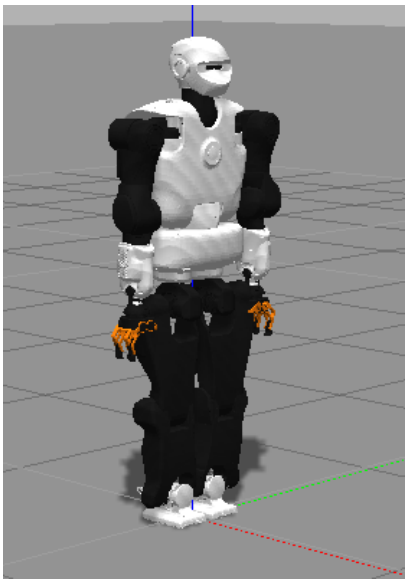


**Figura 4.3:** Robot TALOS en gravedad 0.

## 4.2. Diseño del robot

El robot TALOS de la figura 4.3 es el robot estándar. No obstante, para este TFM se modificará, ya que hay partes que o no son necesarias o son preciso modificar. La ventaja de realizar un robot con forma humana son que pueden compartir el mismo entorno que los humanos, además de poder emplear las mismas herramientas que estos últimos y tener una interacción más natural con ellos [29].

Así pues, no es necesario contar ni con los pies ni con las piernas, ya que el robot se encontrará en situaciones de ingravidez en el espacio. Es por ello por lo que se eliminan, quedando el robot de la figura 4.4.

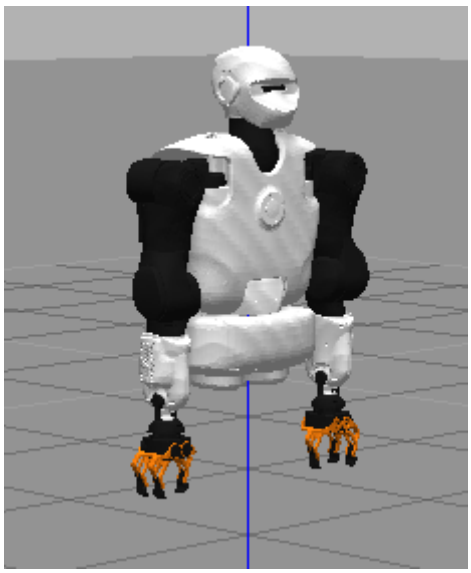


**Figura 4.4:** Robot TALOS despernado en gravedad 0.

Una vez se tiene el robot sin piernas, es el momento de continuar con el ajuste del mismo para la tarea que se desea realizar. Así pues, se cambian las pinzas con las que cuenta el robot de la figura 4.4 por unas manos robóticas. Estas manos, llamadas hey5, se tomarán prestadas del robot TIAGO. Estas pueden verse en la figura 4.5. El resultado de eliminar las pinzas del robot y sustituirlas por dichas manos puede verse en la figura 4.6.


**Figura 4.5:** Robot REEM-C, del cual se tomarán las manos del mismo para incorporarlas al robot TALOS. Fuente: http://erl.pal-robotics.com/

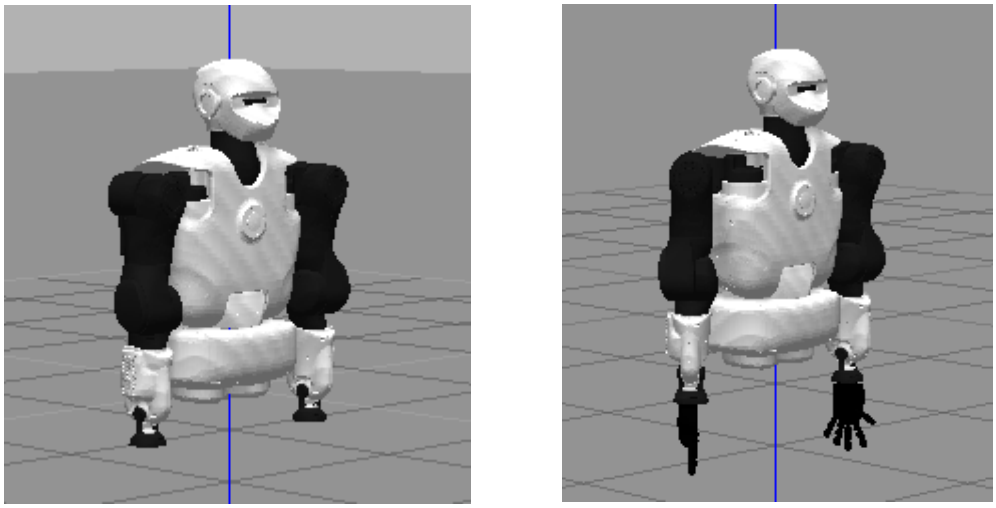


**Figura 4.6:** Robot TALOS sin pinzas (izquierda) y robot TALOS con las manos del robot TIAGO.

Ahora que se ha obtenido la estructura del robot que se empleará de aquí en adelante, ha llegado el momento de personalizar los colores de este. Como se ha realizado este TFM con un profesor asociado al grupo de investigación HURO, se han escogido los colores que emplean en su logo. Este logo puede verse en la figura 4.7, mientras que el robot resultante puede verse en la figura 4.8.

**Figura 4.7:** Logo del grupo Human Robotics (HuRo) de la Universidad de Alicante. Fuente: http://www.huro.ua.es/.

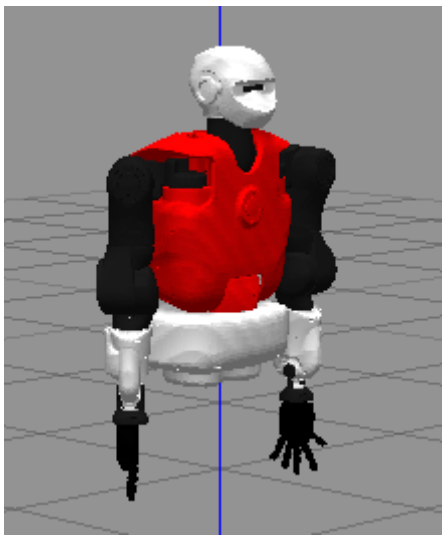


**Figura 4.8:** Robot TALOS con los colores del grupo HuRo.

Por último, se establecerá el robot diseñado junto a un modelo de la ISS en gazebo, de forma que se tiene un escenario en simulación pero realista de la aplicación que se desea desarrollar. Tanto robot como ISS pueden verse en las figuras 4.9 y 4.10. En la figura 4.9 se muestra la ISS de forma completa, en la que el robot creado puede verse como un pequeño punto de color rojo, situado en la parte inferior central. En la figura 4.10 se ha hecho zoom sobre el robot, de forma que se ve a este y una pequeña sección de la ISS.

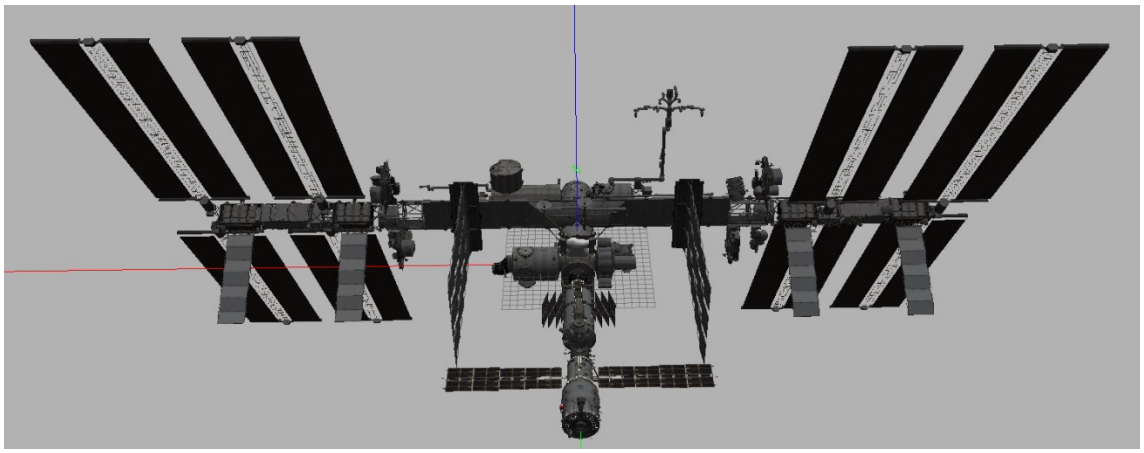


**Figura 4.9:** ISS y robot TALOS en gazebo.

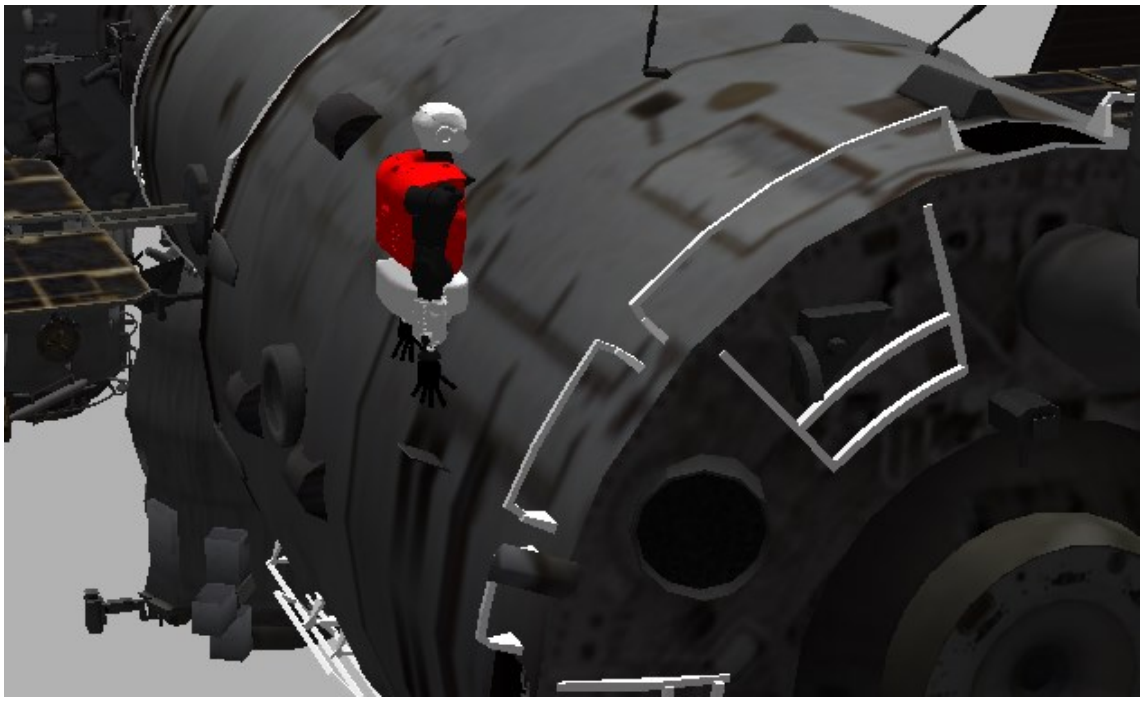


**Figura 4.10:** Robot creado cerca de la ISS.

## 4.3. Control del robot

El control del robot surge de la necesidad de controlar el movimiento. Se trata de una tecnología oculta, la cual mejora las prestaciones del sistema. Únicamente se nota su presencia cuando se produce un fallo (quedando el sistema sin control). No obstante, es imprescindible para asegurar un funcionamiento correcto de cualquier sistema biológico o creado por el ser humano.

En la jerga de los ingenieros de control, la $planta\ o\ sistema$ es el conjunto de elementos físicos relacionados entre sí de forma que, si se realiza una modificación en alguno de ellos, se influye en el resto. Se pretende encontrar una manera de imponer a la salida $y$ el comportamiento deseado, dada una referencia $r$, que es el comportamiento deseado, actuando de una forma factible sobre la planta mediante la señal $u$. Este control debe realizarse a pesar de la presencia de $perturbaciones\ externas\ o\ w$, efectos dinámicos no modelados, errores iniciales respecto a la tarea deseada e incertidumbre en los parámetros del robot o en el proceso.

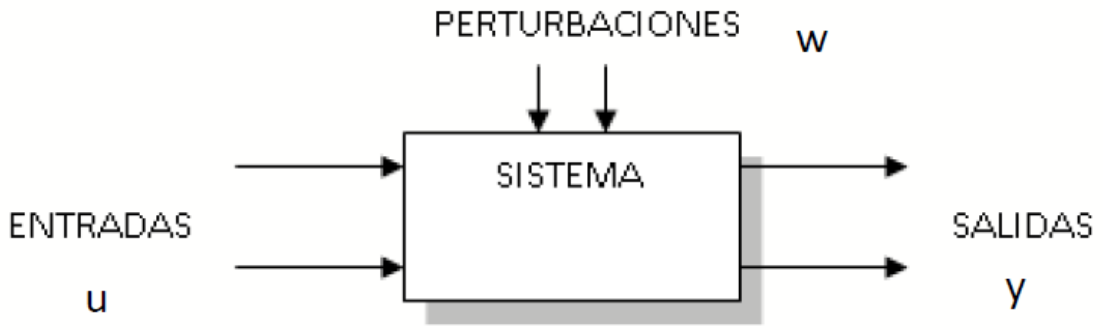



**Figura 4.11:** Esquema de un sistema de control genérico. Fuente: [30]

Así pues, surgen dos métodos diferentes de realizar dicho control: referencia constante (regular la posición / orientación del extremo o de la configuración articular) o referencia variante en el tiempo (seguimiento de una trayectoria).

El sistema puede controlarse en bucle abierto, que es cuando la señal de entrada actúa de forma directa sobre el controlador, o en bucle cerrado, que es una prolongación del bucle abierto en el que se toma una medida a la salida y se le pasa a la entrada. Esto se consigue incluyendo una realimentación interna al sistema, lo que permite que sea insensible a perturbaciones y errores en el modelo [30].

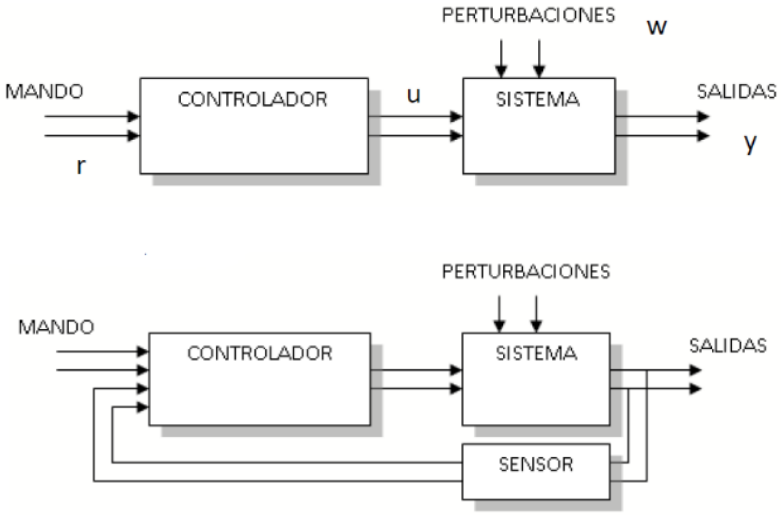



**Figura 4.12:** Sistema en bucle abierto (arriba) y en bucle cerrado (abajo). Fuente: [30]

En los siguientes apartados se estudiará el control multiarticular y el control cartesiano del robot. El primero de ellos consiste en controlar varias articulaciones a la vez, proporcionando un valor a cada una en grados o radianes (empleando la cinemática directa para alcanzar el objetivo), mientras que el segundo consiste en definir un punto del espacio al robot y que el decida como llegar al mismo (cinemática inversa) [31].

### 4.3.1. Controlador multiarticular

Dado un robot con n articulaciones $q = [q_1, q_2, \ldots, q_n]$, un controlador multiarticular es capaz de controlar todas ellas a la vez. De esta manera, se genera el par articular para enviar a cada actuador del robot, además de tener en cuenta el movimiento del resto de articulaciones. En general, estos controladores presentan las siguientes características:

- Un robot manipulador ha de compensar fuerzas externas (gravedad o inercia) para garantizar el correcto seguimiento de las trayectorias durante el movimiento. Sin embargo, el modelo dinámico de un manipulador puede proporcionar información acerca de estas fuerzas externas, permitiendo obtener un mejor comportamiento (al poder predecir estas "fuerzas externas").
- La aproximación más empleada actualmente es la de control monoarticular. Este tipo de control no tiene en cuenta el acoplamiento entre las articulaciones. Dicho de otro modo, cuando un robot manipulador se encuentra en movimiento, aparecen, por ejemplo, fuerzas de Coriolis. Las fuerzas de Coriolis han de ser compensadas por cada uno de los controladores de forma independiente. Sin embargo, a partir del modelo dinámico también puede extraerse información de dichas fuerzas e incluirlas en el propio controlador.
- En los controladores multiarticulares la posición a alcanzar o la trayectoria deseada a realizar será la referencia del controlador y la acción de control generará directamente los pares articulares a aplicar a todos y cada uno de los actuadores del robot.

Por lo tanto, los controladores multiarticulares surgen con el objetivo de mejorar el desempeño de los controladores monoarticulares cuando el robot describe trayectorias rápidas o con gran precisión. En estos casos, los efectos de acoplamiento entre las articulaciones de un robot son más notables y se hace necesario definir un controlador multivariable que genere directamente los pares para todas las articulaciones teniendo en cuenta los posibles pares de acoplamiento entre las articulaciones.

De forma genérica, el modelo dinámico de un robot puede expresarse como en (4.4).

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = \tau \tag{4.4}$$

En ella pueden verse los términos:

- $M\ (q)$ : matriz de inercia simétrica definida positiva.
- $\ddot{q}$ : aceleraciones articulares.
- $C\ (q, \dot{q})$ : vector de fuerzas centrípetas y de Coriolis.
- $\dot{q}$ : velocidades articulares.
- $g\ (q)$ : vector de fuerzas gravitacionales.
- $q$ : posiciones articulares.
- $\tau$ : vector de pares articulares.

Este modelo dinámico puede expresarse en términos de variables de estado, tomando la posición y velocidad articular (siendo estas $[\mathrm{q}^{\mathrm{T}} \quad \dot{\mathrm{q}}^{\mathrm{T}}]^{\mathrm{T}}$), quedando (4.5).

$$\frac{\delta}{\delta t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M\ (q)^{-1}\ [\tau\ (t)\ -\ C\ (q, \dot{q})\ \dot{q}\ -\ g\ (q)] \end{bmatrix} \tag{4.5}$$

El objetivo del control de posición (4.6) es determinar los pares articulares necesarios para llevar al robot a la posición deseada.

$$\lim_{t \to \infty} \tilde{q}\ (t)\ =\ 0 \tag{4.6}$$

En ella puede verse el término:

- $\tilde{q}$ : error de posición articular, equivalente a $q_d - q$.

Así pues, el control genérico de un robot quedaría tal y como puede verse en la figura 4.13 [32].

$$\boldsymbol{\tau = \tau(q, \dot{q}, \ddot{q}, q_d, M(q), C(q, \dot{q}), g(q))}$$

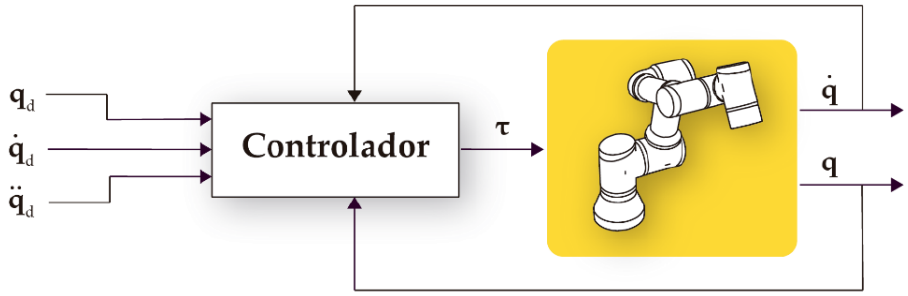



**Figura 4.13:** Esquema genérico de control de un robot. Fuente: [32]

#### 4.3.1.1. Controlador P

La acción de control de un regulador P se corresponde con (4.7). Un esquema de este puede verse en la figura 4.14.

$$K_p\tilde{q} = \tau \tag{4.7}$$

En ella puede verse el término nuevo:

- $K_p$ : matriz definida positiva de constantes proporcionales.

El comportamiento en bucle cerrado se obtiene igualando (4.4) y (4.7), obteniendo:

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = K_p\tilde{q} \tag{4.8}$$

El comportamiento en bucle cerrado, pero en representación de estados, asumiendo $q_d$ constante y despejando la aceleración articular de (4.8), queda (4.9) [32].

$$\frac{\delta}{\delta t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q_d - \tilde{q})^{-1}\left[K_p\tilde{q} - C(q_d - \tilde{q},\dot{q})\,\dot{q} - g(q_d - \tilde{q})\right] \end{bmatrix} \tag{4.9}$$

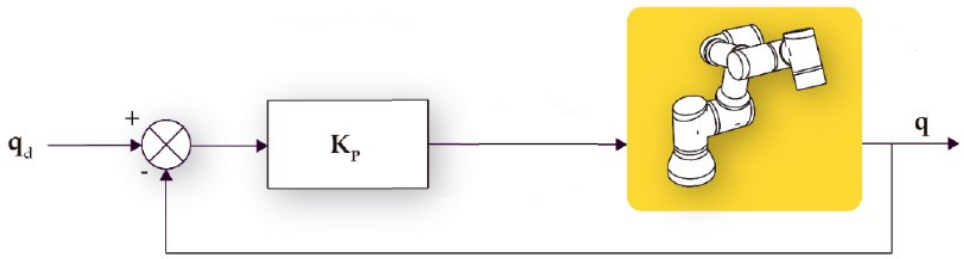



**Figura 4.14:** Controlador P de posición multiarticular. Fuente: [32]

#### 4.3.1.2. Controlador PD

La acción de control de un regulador PD se corresponde con (4.10). Un esquema de este puede verse en la figura 4.15.

$$K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} = \tau \tag{4.10}$$

En ella pueden verse nuevos términos:

- $K_v$ : matriz definida positiva de constantes derivativas.
- $\frac{\delta\tilde{q}}{\delta t}$ : derivada con respecto al tiempo del error articular.

El comportamiento en bucle cerrado se obtiene igualando (4.4) y (4.10), quedando (4.11).

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} \tag{4.11}$$

El comportamiento en bucle cerrado, pero en representación de estados, asumiendo $q_d$ constante y despejando la aceleración articular de (4.11), quedaría (4.12) [32].

$$\frac{\delta}{\delta t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q_d - \tilde{q})^{-1}\left[K_p\tilde{q} - K_v\frac{\delta q}{\delta t} - C(q_d - \tilde{q},\dot{q})\,\dot{q} - g(q_d - \tilde{q})\right] \end{bmatrix} \tag{4.12}$$

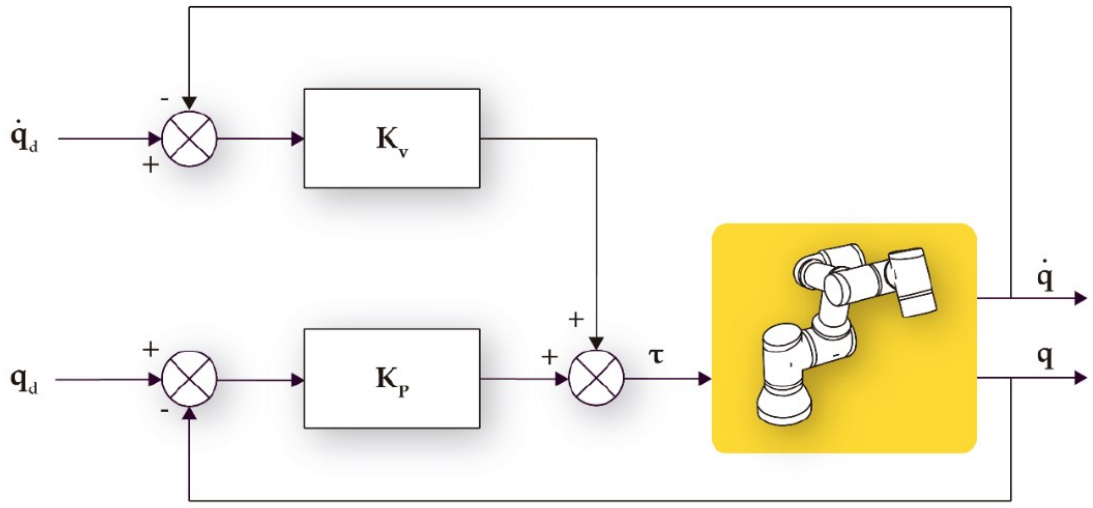



**Figura 4.15:** Controlador PD de posición multiarticular. Fuente: [32]

#### 4.3.1.3. Controlador PD con compensación de gravedad

La acción de control de un regulador PD con compensación de gravedad se corresponde con (4.13). Un esquema de este puede verse en la figura 4.16.

$$K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} + g\,(q) = \tau \tag{4.13}$$

En ella pueden verse nuevos términos:

- $g\,(q)$ : componente de gravedad del modelo dinámico del robot.

El comportamiento en bucle cerrado se obtiene igualando (4.4) y (4.13), obteniendo (4.14).

$$M\,(q)\,\ddot{q} + C\,(q,\dot{q})\,\dot{q} + g(q) = K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} + g\,(q) \tag{4.14}$$

El comportamiento en bucle cerrado, pero en representación de estados, asumiendo $q_d$ constante y despejando la aceleración articular de (4.14), quedaría (4.15) [32].

$$\frac{\delta}{\delta t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M\,(q_d - \tilde{q})^{-1}\left[K_p\tilde{q} - K_v\frac{\delta q}{\delta t} - C\,(q_d - \tilde{q},\dot{q})\,\dot{q}\right] \end{bmatrix} \tag{4.15}$$

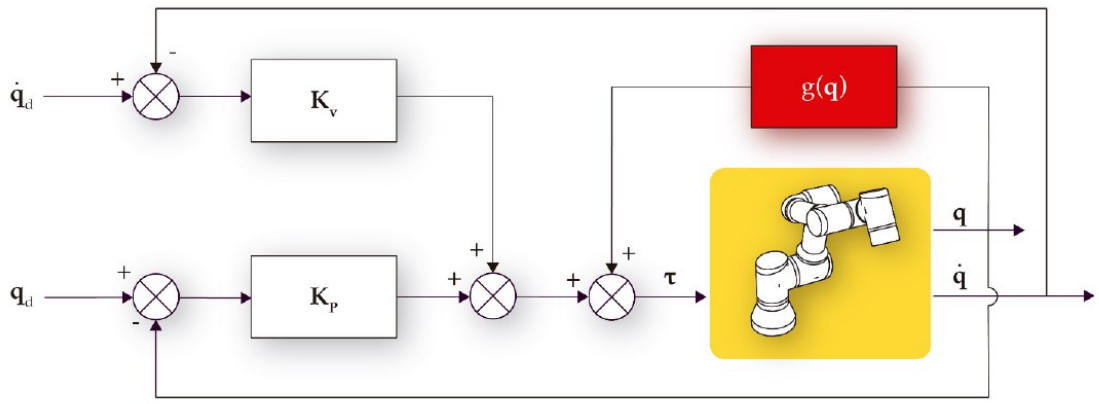



**Figura 4.16:** Controlador PD con compensación de gravedad de posición multiarticular. Fuente: [32]

#### 4.3.1.4. Controlador PD con compensación precalculada de gravedad

La acción de control de un regulador PD con compensación precalculada de gravedad se corresponde con (4.16). Un esquema de este puede verse en la figura 4.17.

$$K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} + g\,(q_d) = \tau \tag{4.16}$$

En ella pueden verse nuevos términos:

- $g\,(q_d)$ : componente de gravedad del modelo dinámico del robot en la posición final a alcanzar.

El comportamiento en bucle cerrado se obtiene igualando la (4.4) y (4.16), obteniendo (4.17).

$$M\,(q)\,\ddot{q} + C\,(q,\dot{q})\,\dot{q} + g(q) = K_p\tilde{q} + K_v\frac{\delta\tilde{q}}{\delta t} + g\,(q_d) \tag{4.17}$$

El comportamiento en bucle cerrado, pero en representación de estados, asumiendo $q_d$ constante y despejando la aceleración articular de (4.17), quedaría (4.18) [32].

$$\frac{\delta}{\delta t}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M\,(q_d - \tilde{q})^{-1}\left[K_p\tilde{q} - K_v\frac{\delta q}{\delta t} - C\,(q_d - \tilde{q},\dot{q})\,\dot{q} + g\,(q_d) - g\,(q_d - \tilde{q})\right] \end{bmatrix} \tag{4.18}$$

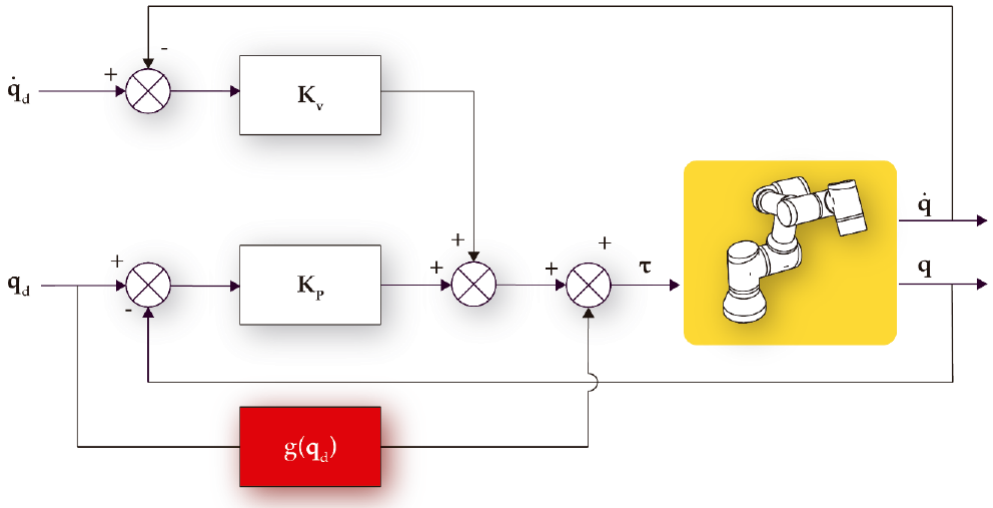



**Figura 4.17:** Controlador PD con compensación precalculada de gravedad de posición multiarticular. Fuente: [32]

#### 4.3.1.5. Controlador PID

La acción de control de un regulador PID se corresponde con (4.19). Un esquema de este puede verse en la figura 4.18.

$$K_p\tilde{q} \;+\; K_v\frac{\delta\tilde{q}}{\delta t} \;+\; K_i\int_0^t \tilde{q}\,(\sigma)\,d\sigma \;=\; \tau \tag{4.19}$$

También puede ser expresada de esta otra forma:

$$\begin{gathered} K_p\tilde{q} \;+\; K_v\frac{\delta\tilde{q}}{\delta t} \;+\; K_i\,\xi \;=\; \tau \\ \dot{\xi} \;=\; \tilde{q} \end{gathered} \tag{4.20}$$

En ella pueden verse nuevos términos:

- $K_i$ : matriz definida positiva de constantes integrales.

El comportamiento en bucle cerrado se obtiene igualando (4.4) y (4.20), obteniendo (4.21).

$$M\,(q)\,\ddot{q} \;+\; C\,(q,\dot{q})\,\dot{q} \;+\; g(q) \;=\; K_p\tilde{q} \;+\; K_v\frac{\delta\tilde{q}}{\delta t} \;+\; K_i\,\xi \tag{4.21}$$

El comportamiento en bucle cerrado, pero en representación de estados, despejando la aceleración articular de (4.21), quedaría (4.22) [32].

$$\frac{\delta}{\delta t}\begin{bmatrix}\xi\\ \tilde{q}\\ \frac{\delta\tilde{q}}{\delta t}\end{bmatrix} = \begin{bmatrix}\tilde{q}\\ \frac{\delta\tilde{q}}{\delta t}\\ -M\,(q_d \;-\; \tilde{q})^{-1}[K_p\tilde{q} \;+\; K_v\frac{\delta q}{\delta t} + K_i\,\xi \;+\; C\,(q_d \;-\; \tilde{q}, -\frac{\delta\tilde{q}}{\delta t})\,\frac{\delta\tilde{q}}{\delta t} \;-\; g\,(q_d \;-\; \tilde{q})]\end{bmatrix} \tag{4.22}$$

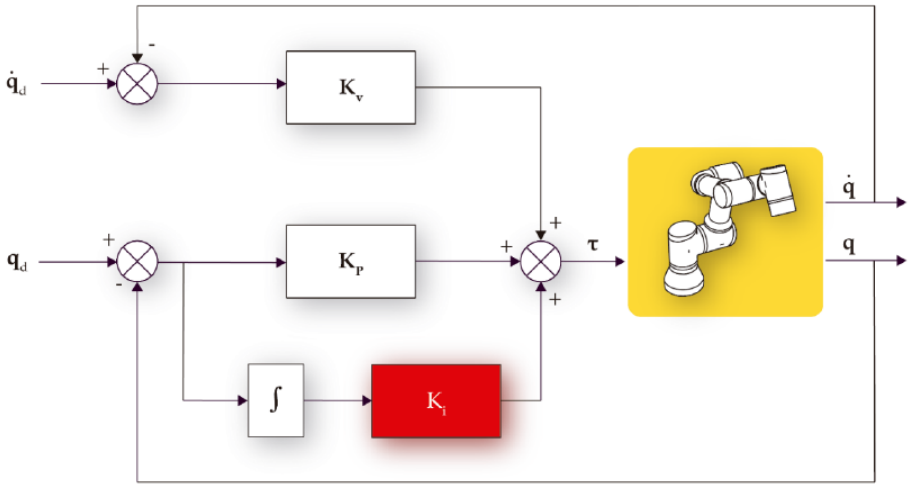



**Figura 4.17:** Controlador PD con compensación precalculada de gravedad de posición multiarticular. Fuente: [32]

### 4.3.2. Controlador cartesiano

Este controlador es un poco diferente respecto a los anteriores. En este, la referencia es la posición y orientación de un punto en el espacio tridimensional (teniéndose 6 coordenadas en total). De esta manera, el robot elige el ángulo que debe girar cada una de las articulaciones para llegar al objetivo.

Para ello hace uso de la matriz jacobiana (4.23). La cinemática diferencial permite estudiar las relaciones entre el espacio articular y el cartesiano, considerando velocidades y posiciones. La matriz jacobiana establece la relación entre las velocidades de las coordenadas articulares y las de posición y orientación del extremo del robot [31].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = \begin{bmatrix} \frac{\delta f_x}{\delta q_1} & \cdots & \frac{\delta f_x}{\delta q_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta f_\gamma}{\delta q_1} & \cdots & \frac{\delta f_\gamma}{\delta q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \tag{4.23}$$

En (4.23) las tres primeras filas se corresponden con la velocidad lineal del extremo, mientras que las tres segundas filas se corresponden con la velocidad angular del extremo.

Además, en el proceso de cálculo, se realizan cambios entre el espacio articular y el cartesiano para poder llevar a cabo el control. Para ello se emplea el modelo cinemático directo y el inverso. De esta manera, al pasar del espacio cartesiano al articular aplicando la cinemática inversa, se obtendrán múltiples soluciones o ninguna, mientras que al pasar del espacio articular al cartesiano aplicando la cinemática directa, se obtendrá una única solución [33].

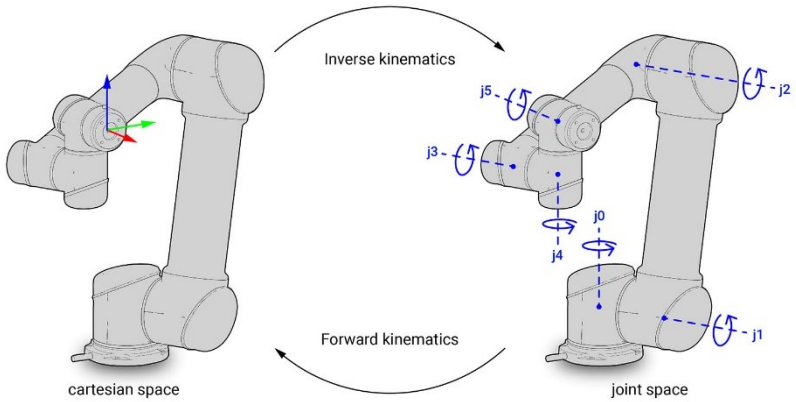



**Figura 4.18:** Relación entre espacio cartesiano, articular y la transformación cinemática a aplicar. Fuente: [34]

Para empezar, se construye una cadena de articulaciones, empleando los nombres de la primera y última articulación a controlar. Esta cadena también contiene un puntero al estado del robot, para ser capaces de acceder al mismo el cada ciclo del bucle de control. Tras ello, se configura KDL. Se deben crear dos solvers, para lo que es necesario conocer la cadena cinemática del robot.

Se sigue calculando la posición cartesiana inicial en la que se encuentra el robot. Para poder calcularla, se lee la posición articular de la cadena creada y se emplea KDL, empleando la cinemática directa.

El controlador de forma continua obtiene, en primer lugar, la posición del extremo del robot. Esto lo consigue leyendo el objeto cadena. Tras ello, calcula la posición del extremo del robot y la matriz jacobiana empleando los solvers de KDL. La velocidad de cada articulación se consigue haciendo el producto de la matriz jacobiana y el error actual.

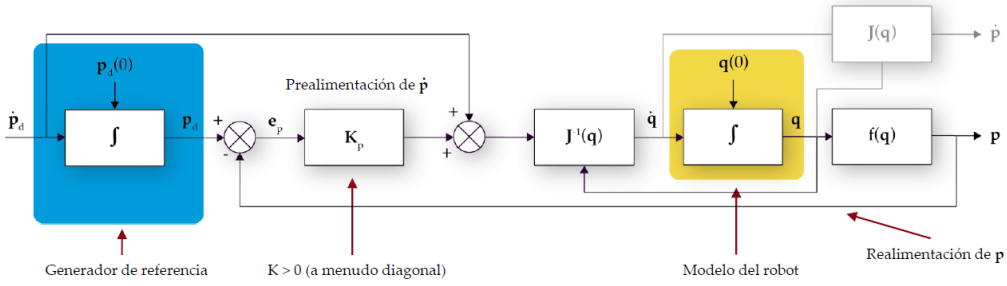



**Figura 4.19:** Controlador cartesiano de posición multiarticular. Fuente: [32]

# 5. Resultados

Se han llevado a cabo una serie de simulaciones con el objetivo de analizar el rendimiento de los controladores creados. Se han simulado distintos movimientos con el objetivo de comprobar si el robot se encuentra bien construido y si es posible realizar el control del mismo.

En los subapartados siguientes se realiza una descripción de los movimientos a realizar, las posiciones deseadas en el espacio cartesiano y articular, las ganancias empleadas y las gráficas de esfuerzo, posición y velocidad obtenidas tras los movimientos.

Se han considerado 3 casos diferentes, los cuales pueden ser usados durante distintas tareas en la ISS:

1. Una maniobra de rotación de la mano.
2. Una maniobra de rotación del codo.
3. Una maniobra de rotación del hombro y codo.

## 5.1. Rotación de la mano

En este movimiento se pretende llevar al robot entre las posiciones inicial y final que pueden verse en la figura 5.1. Este movimiento puede entenderse como una rotación de 90 grados de la tercera articulación del brazo robótico.

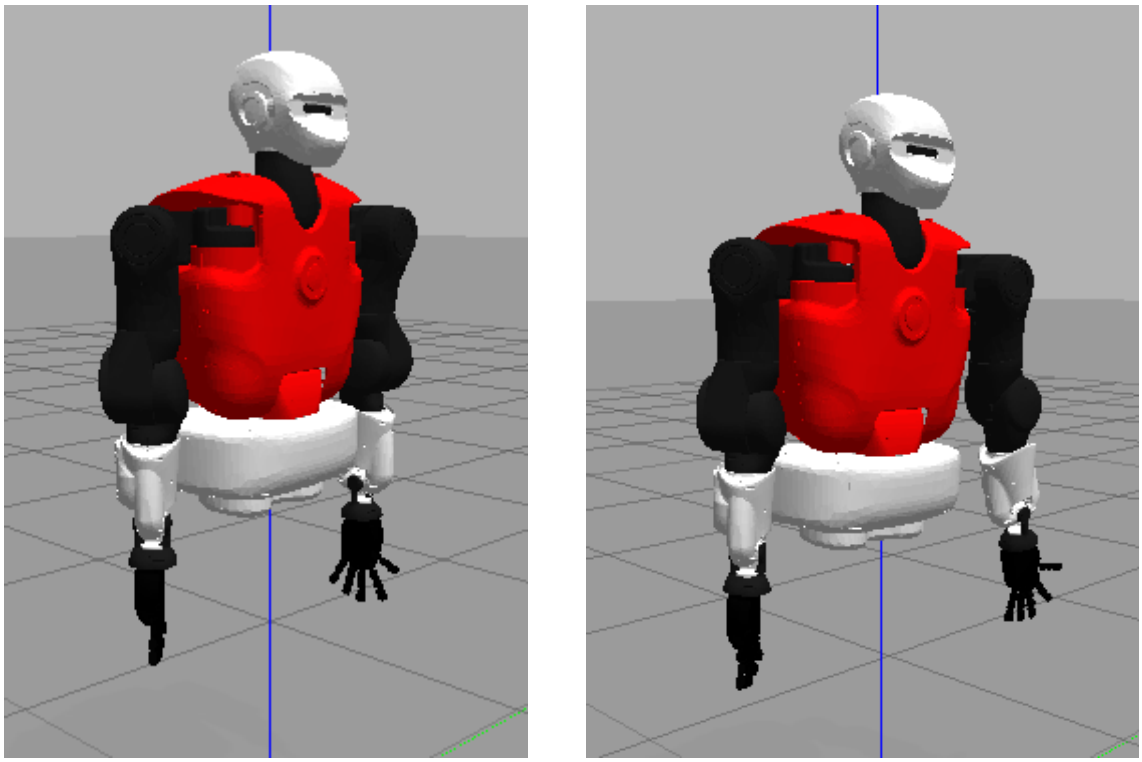


**Figura 5.1:** Posición inicial (izquierda) y final (derecha) del movimiento de rotación de la mano.

En caso de proporcionar una posición articular como objetivo, se pueden emplear los controladores P multiarticular, PD multiarticular, PD con compensación de gravedad multiarticular, PD con compensación precalculada de gravedad multiarticular y PID multiarticular. El objetivo del movimiento vendría dado por (5.1).

$$X_d = [0\ 0\ 1.57\ 0\ 0\ 0\ 0] \tag{5.1}$$

En ella puede verse el término:

- $X_d$ : posición deseada a alcanzar por el controlador.

El mejor resultado de todos los controladores mencionados fue el obtenido con el controlador P. No obstante, el resto de las gráficas eran similares a las del controlador P, aunque estas presentan un menor tiempo de establecimiento y amplitud de las señales. Para ello, se han empleado las ganancias de (5.2).

$$K_p = [2.5\ 0.05\ 5\ 7.5\ 50\ 50\ 0.02] \tag{5.2}$$

Los resultados obtenidos en la figura 5.2 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, se consigue la estabilidad del sistema apenas transcurridos 0.75 segundos, lo cual es muy rápido.

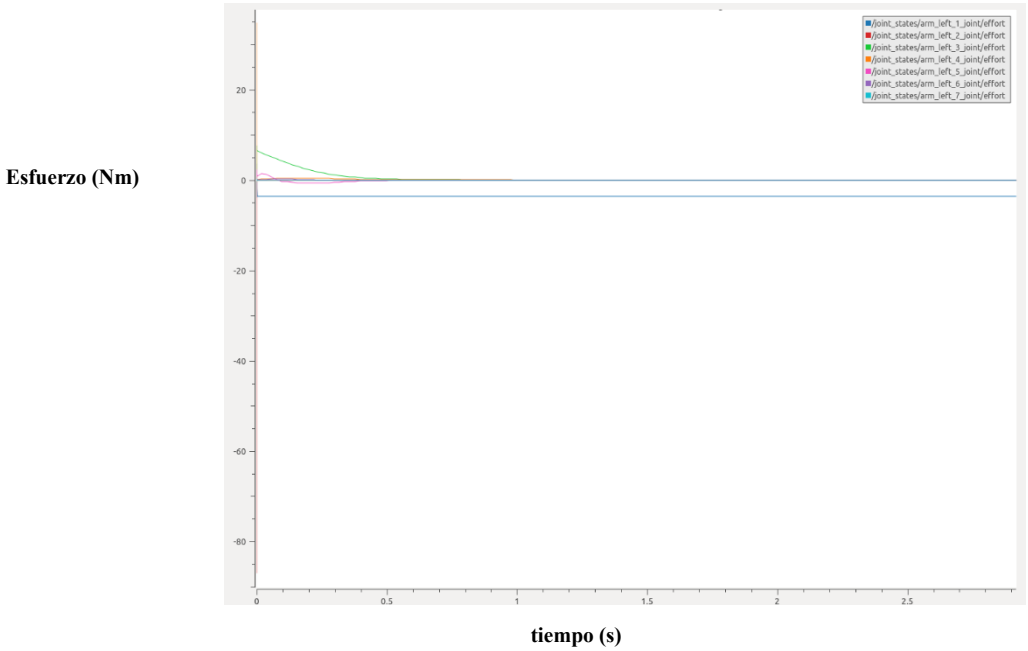



**Figura 5.2:** Esfuerzo realizado por el brazo izquierdo con el controlador P para realizar la rotación de la mano.

En la figura 5.3 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, la articulación 3 alcanza su objetivo cuando apenas ha transcurrido 1 segundo desde el comienzo del movimiento. No obstante, las articulaciones 1 y 2 obtienen posiciones finales diferentes a las deseadas. Esto es debido a que, para alcanzar el objetivo con la articulación 3, las posiciones de estas articulaciones son modificadas. El error no es muy grande, si se tiene en cuenta de que se está trabajando con un controlador P.

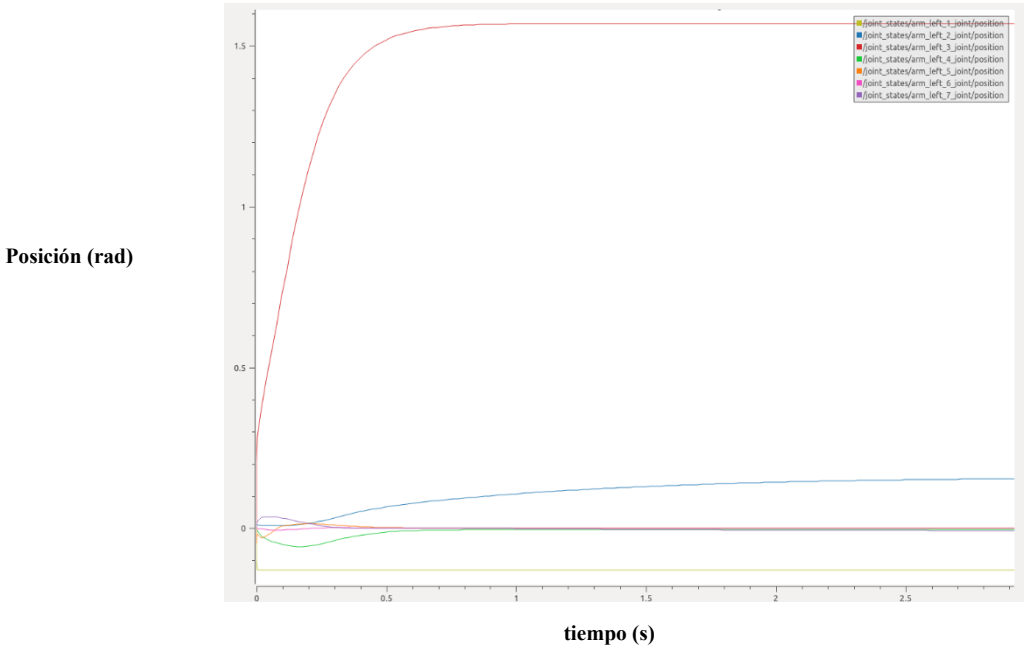



**Figura 5.3:** Posición realizado por el brazo izquierdo con el controlador P para realizar la rotación de la mano.

En la figura 5.4 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, se estabilizan en 0, aunque surgen unas periodicidades a lo largo del tiempo (aunque son pequeñas). Estas son debidas a que, el controlador realiza un esfuerzo, se pasa de la posición, hace el opuesto, se pasa, etc. Esto sucede porque no hay gravedad en las simulaciones, siendo un hecho curioso y que podría ser estudiado aparte.

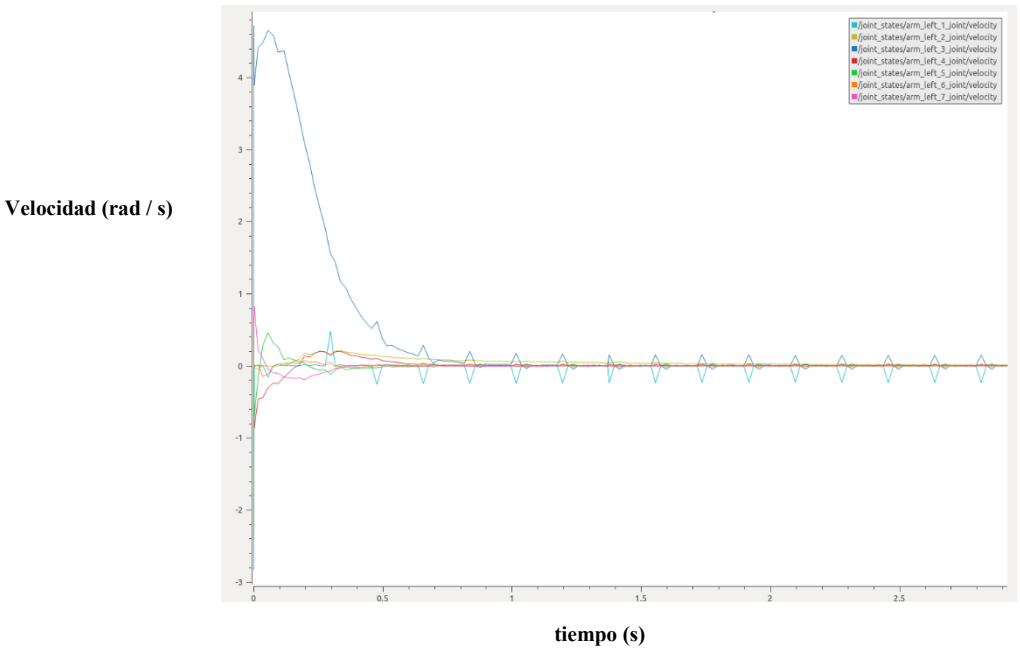



**Figura 5.4:** Velocidades realizadas por el brazo izquierdo con el controlador P para realizar la rotación de la mano.

En caso de proporcionar una posición cartesiana como objetivo, se puede emplear el controlador cartesiano. El objetivo del movimiento vendría dado por (5.1).

$$\begin{aligned} P_d &= [0.2393\ 0.1337\ -0.2066] \\ \theta_d &= [-0.6703427\ 82.9213784\ 0.1997149] \end{aligned} \tag{5.3}$$

En ella pueden verse términos nuevos:

- $\theta_d$ : posición angular deseada a alcanzar por el controlador.

Los resultados obtenidos en la figura 5.5 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, se consigue la estabilidad del sistema, pero es necesario un tiempo mayor para lograrlo (1.5 segundos).

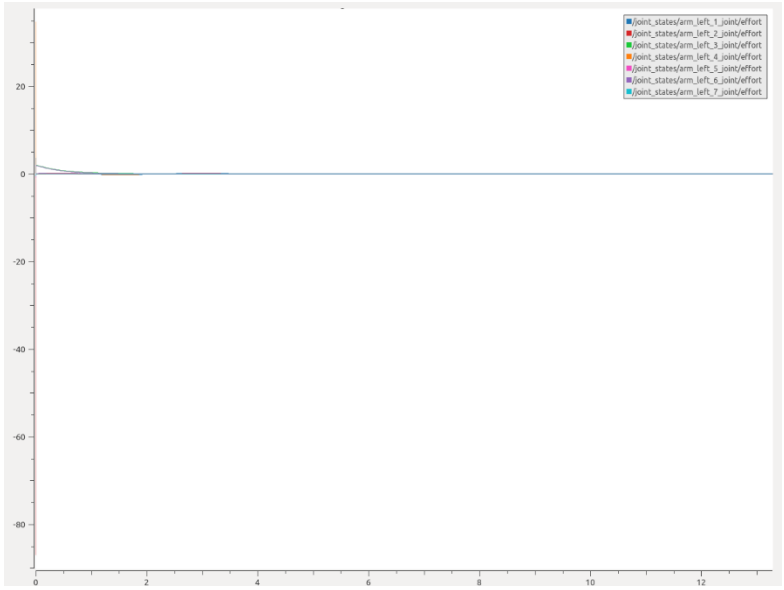



**Figura 5.5:** Esfuerzo realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación de la mano.

En la figura 5.6 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, es necesario mucho más tiempo para conseguir que las articulaciones vayan por el buen camino (hacia el objetivo marcado como posición deseada)

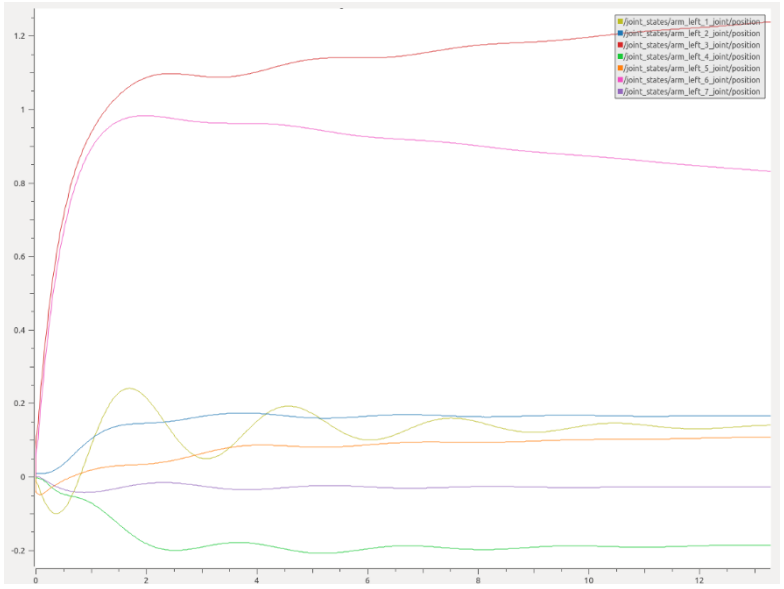



**Figura 5.6:** Posición realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación de la mano.

En la figura 5.7 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, además de necesitar más tiempo para estabilizarse en 0, es necesario que se apliquen velocidades mayores y sucesivas en comparación con el controlador P para el mismo movimiento.

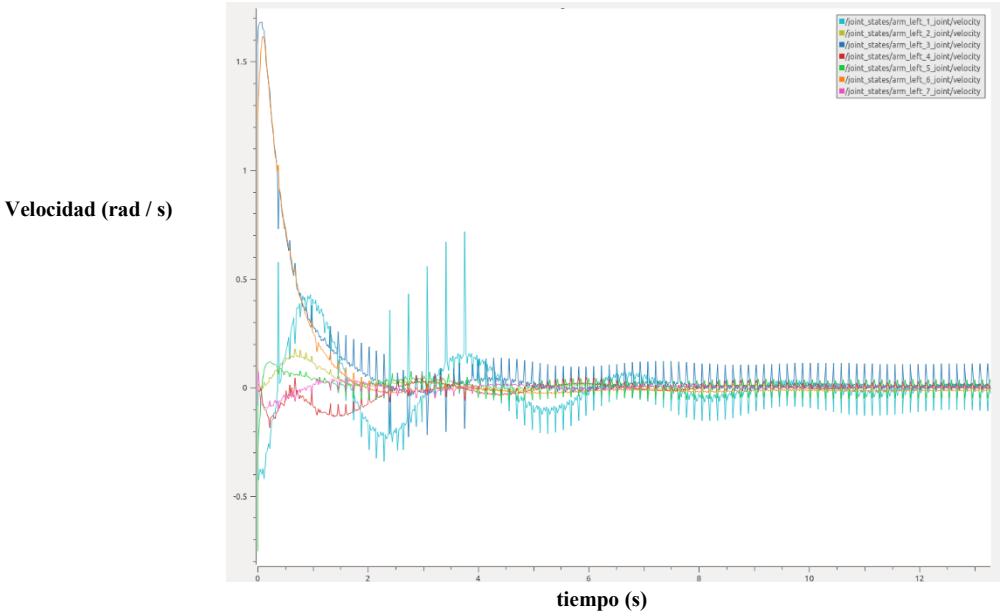



**Figura 5.7:** Velocidades realizadas por el brazo izquierdo con el controlador cartesiano para realizar la rotación de la mano.

## 5.2. Rotación del codo

En este movimiento se pretende llevar al robot entre las posiciones inicial y final que pueden verse en la figura 5.8. Este movimiento puede entenderse como una rotación de 90 grados negativos de la cuarta articulación del brazo robótico.

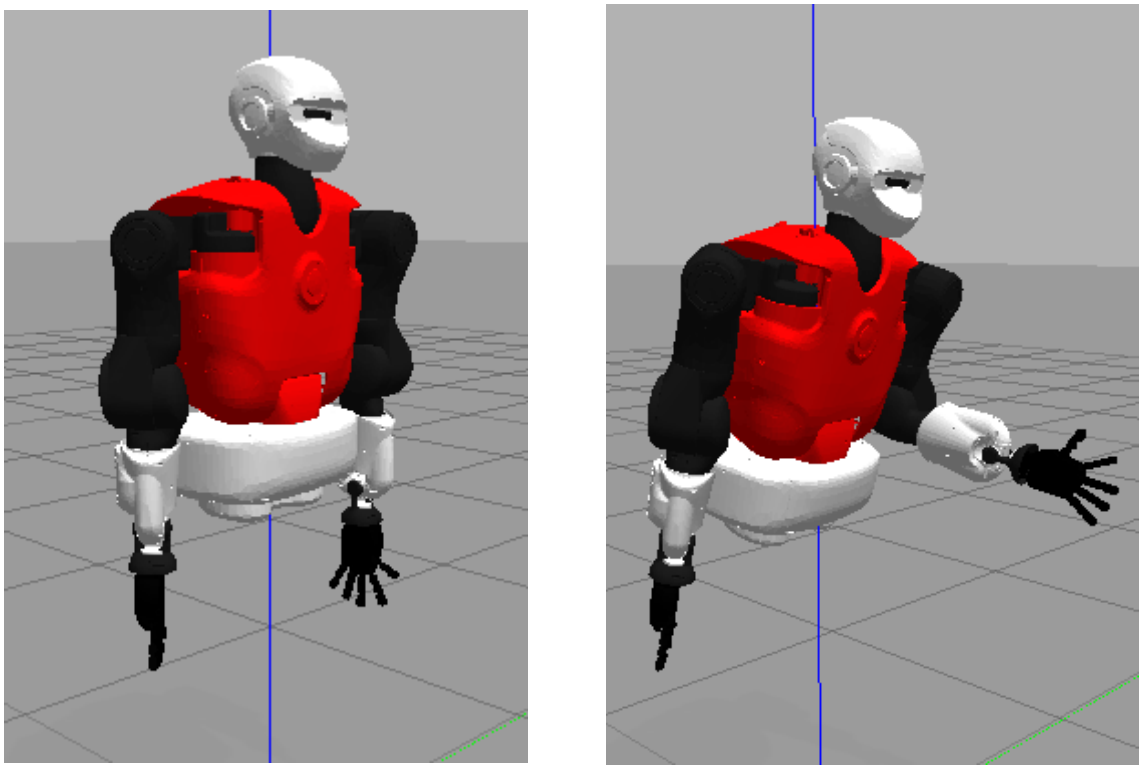


**Figura 5.8:** Posición inicial (izquierda) y final (derecha) del movimiento de rotación del codo.

En caso de proporcionar una posición articular como objetivo, se pueden emplear los controladores P multiarticular, PD multiarticular, PD con compensación de gravedad multiarticular, PD con compensación precalculada de gravedad multiarticular y PID multiarticular. El objetivo del movimiento vendría dado por (5.4).

$$X_d = [0\ 0\ 0\ -1.57\ 0\ 0\ 0] \tag{5.4}$$

El mejor resultado de todos los controladores mencionados fue el obtenido con el controlador P. No obstante, el resto de las gráficas eran similares a las del controlador P, aunque las de este presentan un menor tiempo de establecimiento y menor amplitud de las señales. Para ello, se han empleado las ganancias de (5.5).

$$K_p = [50\ 1\ 50\ 7.5\ 50\ 50\ 0.02] \tag{5.5}$$

Los resultados obtenidos en la figura 5.9 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, es necesario que transcurran 3 segundos para alcanzar la estabilidad.

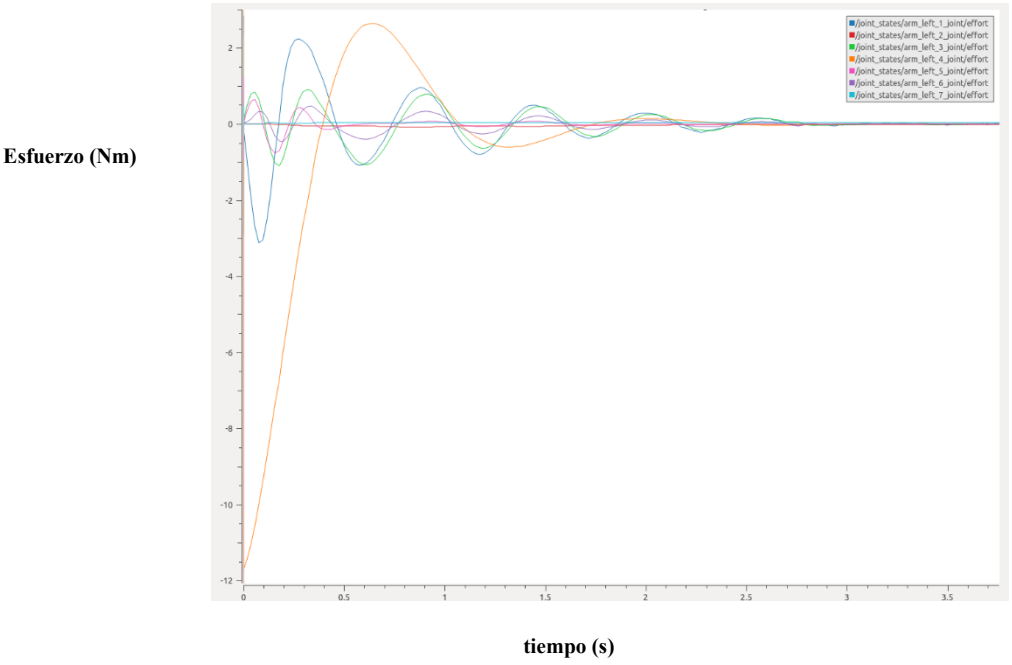



**Figura 5.9:** Esfuerzo realizado por el brazo izquierdo con el controlador P para realizar la rotación del codo.

En la figura 5.10 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, la articulación 7 va un poco a su rollo cuando han transcurrido 3.5 segundos. No obstante, la articulación se queda cerca del 0 en todas las articulaciones a excepción de la 4, que debe alcanzar -1.57 como objetivo, lo cual coincide con la posición objetivo.

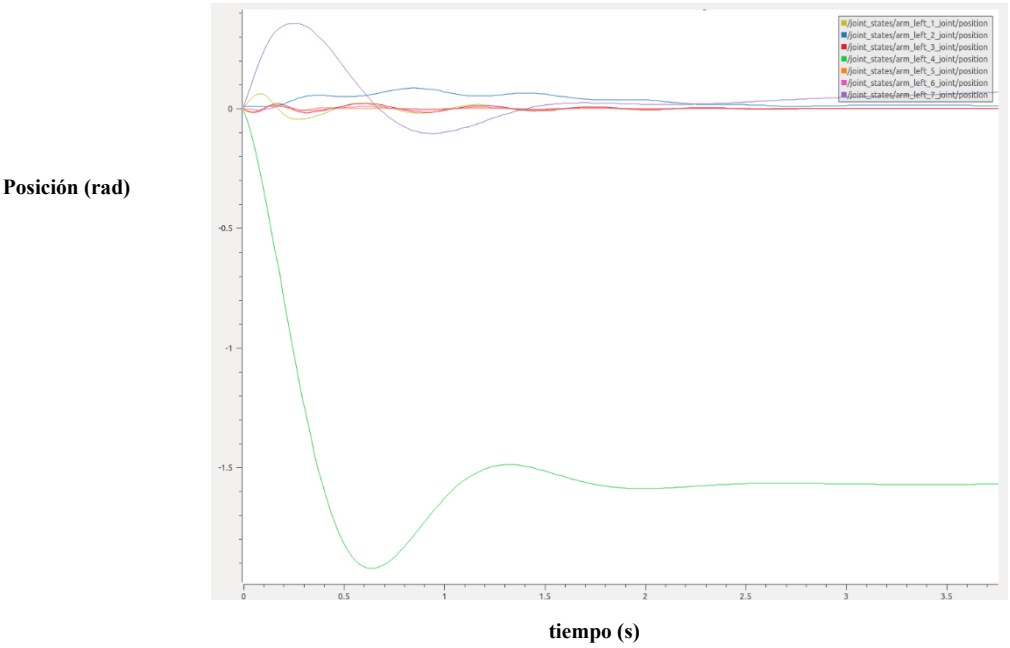



**Figura 5.10:** Posición realizado por el brazo izquierdo con el controlador P para realizar la rotación del codo.

En la figura 5.11 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, se estabilizan en 0, aunque surgen unas periodicidades a lo largo del tiempo (aunque son pequeñas).

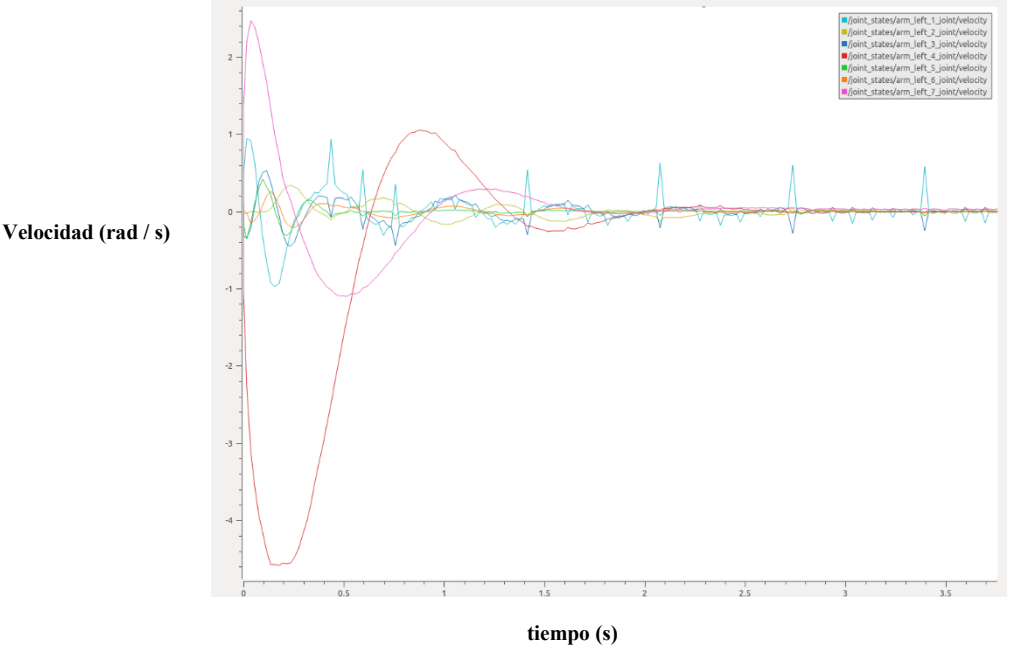



**Figura 5.11:** Velocidades realizadas por el brazo izquierdo con el controlador P para realizar la rotación del codo.

En caso de proporcionar una posición cartesiana como objetivo, se puede emplear el controlador cartesiano. El objetivo del movimiento vendría dado por (5.6).

$$\begin{aligned} P_d &= [0.0049\ 0.1364\ -0.49057] \\ \theta_d &= [-0.088\ -0.0911\ -89.4496] \end{aligned} \tag{5.6}$$

Los resultados obtenidos en la figura 5.12 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, se consigue la estabilidad del sistema, pero es necesario un tiempo mayor para lograrlo (4.5 segundos).

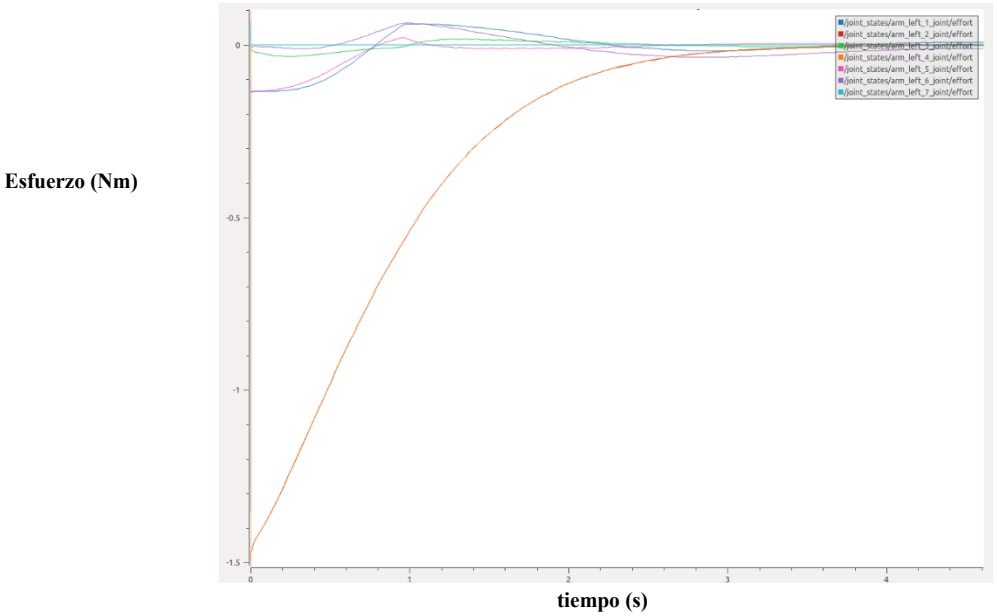



**Figura 5.12:** Esfuerzo realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación del codo.

En la figura 5.13 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, es necesario mucho más tiempo para conseguir que las articulaciones vayan por el buen camino. Sin embargo, las articulaciones 1 y 5 parecen seguir otro patrón de posiciones, diferente al que deberían de llevar.

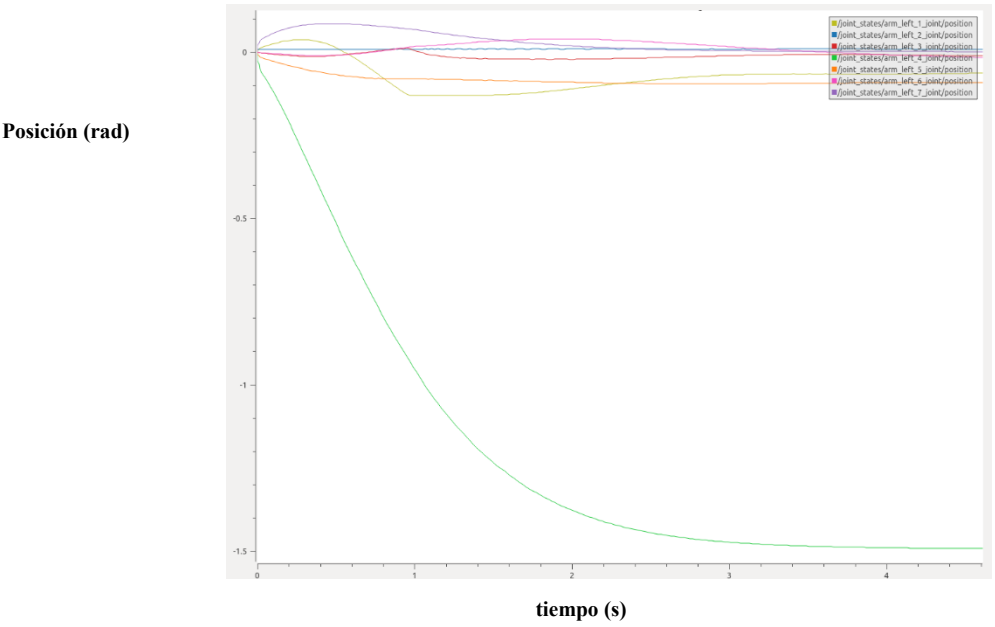



**Figura 5.13:** Posición realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación del codo.

En la figura 5.14 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, además de necesitar más tiempo para estabilizarse en 0, aparecen estas estructuras periódicas nuevamente.

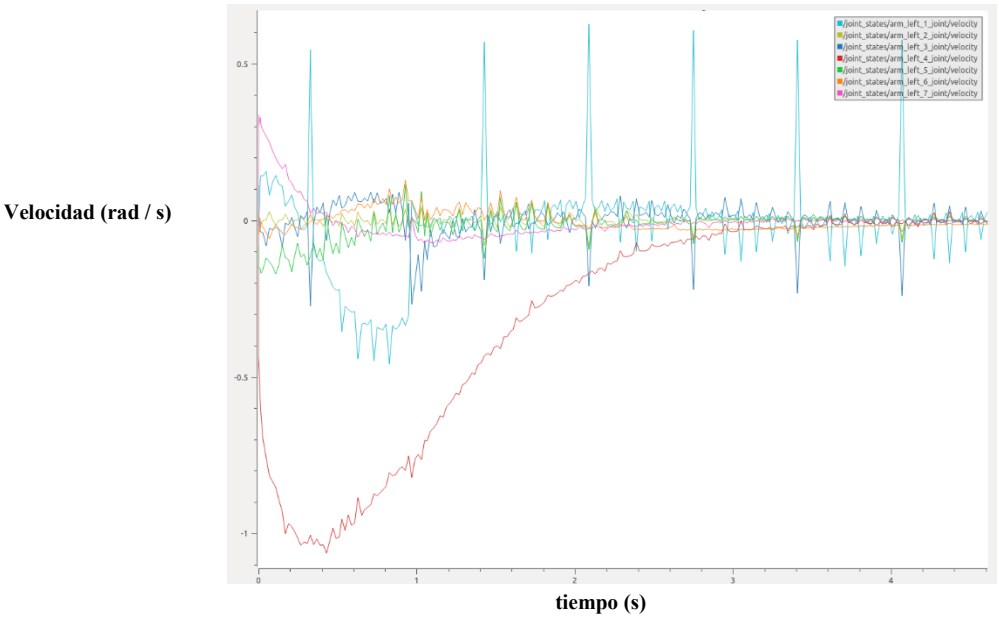



**Figura 5.14:** Velocidades realizadas por el brazo izquierdo con el controlador cartesiano para realizar la rotación del codo.

## 5.3. Rotación del hombro y mano

En este movimiento se pretende llevar al robot entre las posiciones inicial y final que pueden verse en la figura 5.15. Este movimiento puede entenderse como una rotación de 90 grados de la segunda y tercera articulación del brazo robótico.

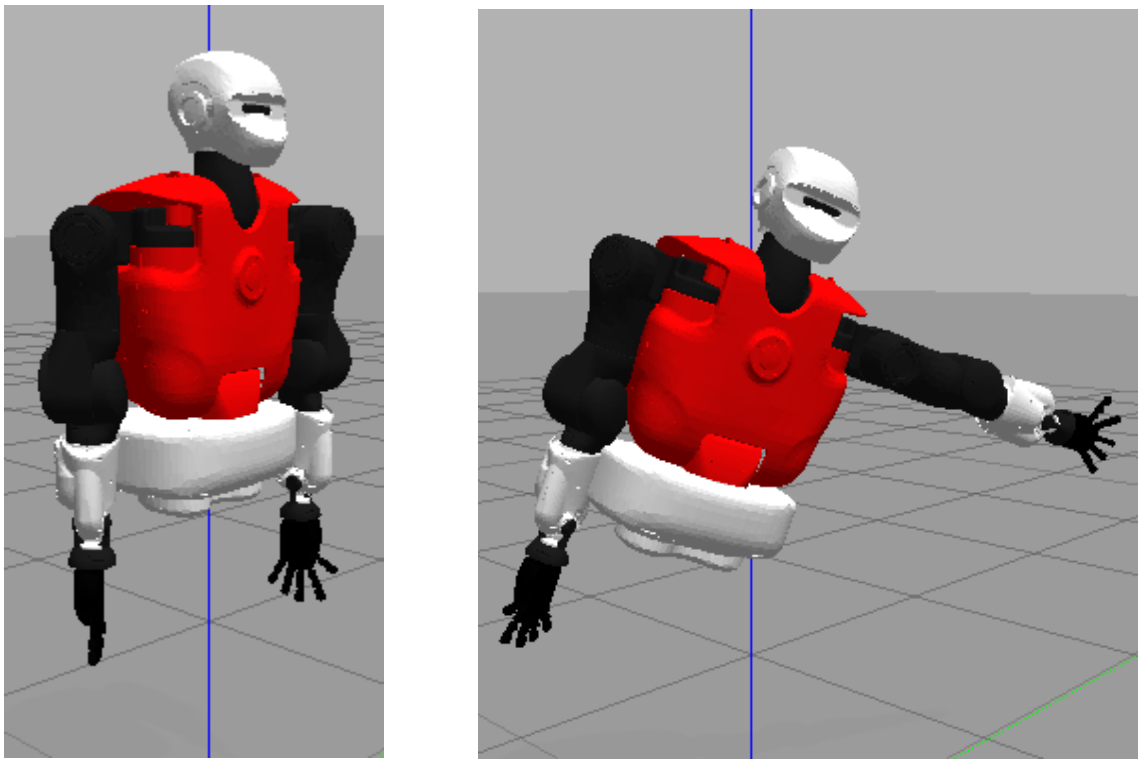


**Figura 5.15:** Posición inicial (izquierda) y final (derecha) del movimiento de rotación de hombro y mano.

En caso de proporcionar una posición articular como objetivo, se puede emplear los controladores P multiarticular, PD multiarticular, PD con compensación de gravedad multiarticular, PD con compensación precalculada de gravedad multiarticular y PID multiarticular. El objetivo del movimiento vendría dado por (5.7).

$$X_d = [0\ 1.57\ 1.57\ 0\ 0\ 0\ 0] \tag{5.7}$$

El mejor resultado de todos los controladores mencionados fue el obtenido con el controlador PID. No obstante, el resto de las gráficas eran similares a las del controlador PID, aunque las de este presentan un menor tiempo de establecimiento y menor amplitud de las señales. Para ello, se han empleado las ganancias de (5.8).

$$\begin{aligned} K_p &= [8\ 2\ 4.5\ 7.5\ 50\ 50\ 0.05] \\ K_v &= [0.2\ 1\ 0.03\ 5\ 0.1\ 5\ 5] \\ K_i &= [1\ 1\ 0.5\ 1\ 1\ 1\ 1] \end{aligned} \tag{5.8}$$

Los resultados obtenidos en la figura 5.16 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, es necesario que transcurra más tiempo para lograr la estabilidad, ya que el movimiento es más amplio que en los ejemplos anteriores.

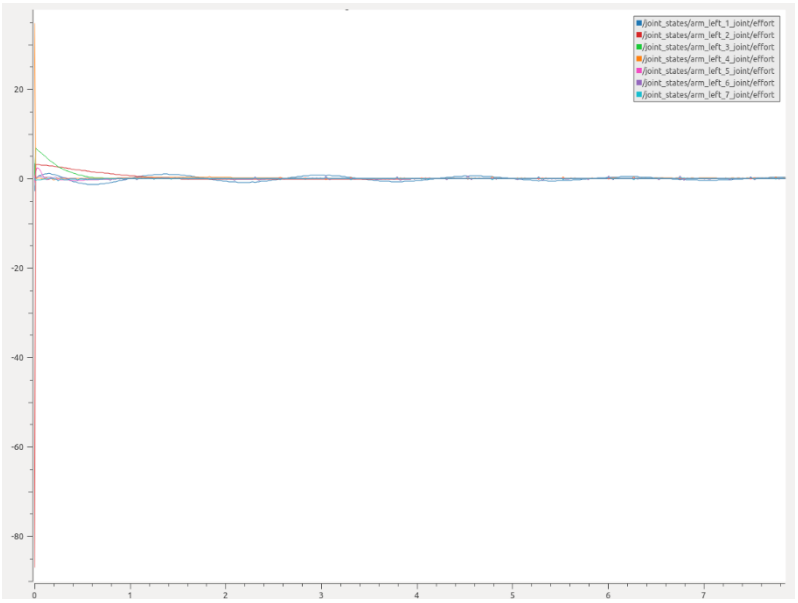



**Figura 5.16:** Esfuerzo realizado por el brazo izquierdo con el controlador PID para realizar la rotación de hombro y codo.

En la figura 5.17 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, la articulación 7, a pesar de ir un poco a su rollo, recupera el buen camino en torno al segundo 7. Las articulaciones 2 y 3 consiguen alcanzar la posición final, siendo la tercera (giro de la mano) la más rápida, siendo esto muy razonable.

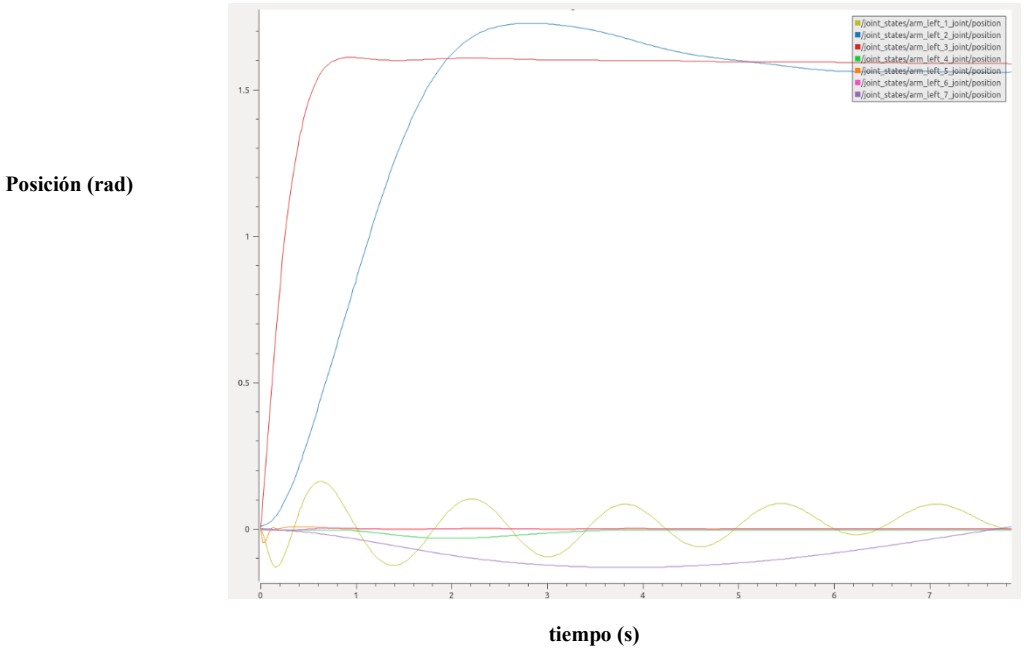



**Figura 5.17:** Posición realizado por el brazo izquierdo con el controlador PID para realizar la rotación de hombro y codo.

En la figura 5.18 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, se estabilizan en 0, aunque surgen unas periodicidades a lo largo del tiempo (aunque son pequeñas).

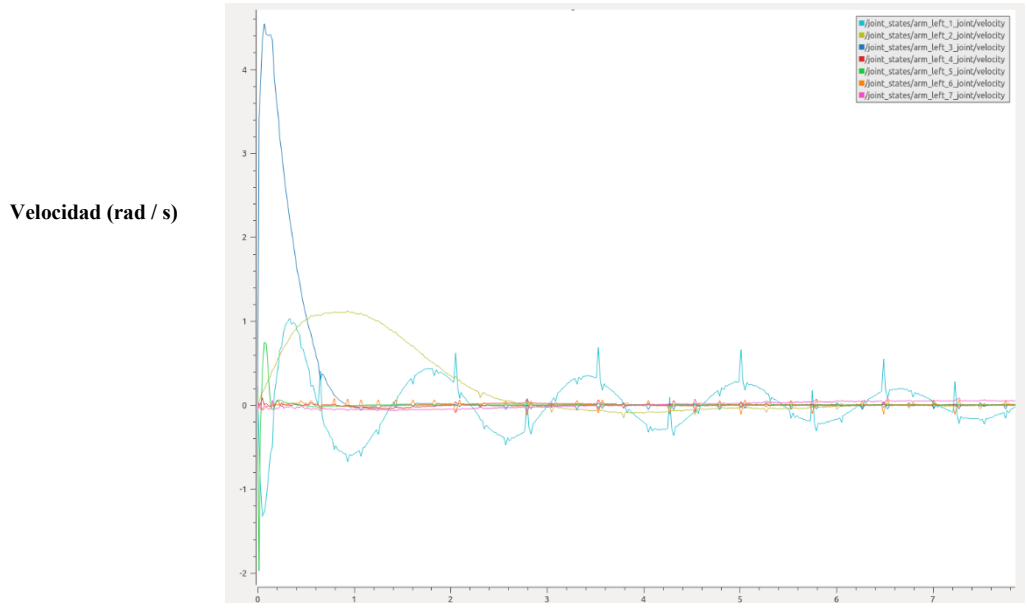



**Figura 5.18:** Velocidades realizadas por el brazo izquierdo con el controlador PID para realizar la rotación de hombro y codo.

En caso de proporcionar una posición cartesiana como objetivo, se puede emplear el controlador cartesiano. El objetivo del movimiento vendría dado por (5.9).

$$P_d = [0.0029 \ -0.10249 \ -0.2065] \\ \theta_d = [0 \ 90 \ 0] \tag{5.9}$$

Los resultados obtenidos en la figura 5.19 muestran los esfuerzos realizados por el controlador para alcanzar la posición final. Como puede verse, se consigue que los controladores no saturen en ningún momento.

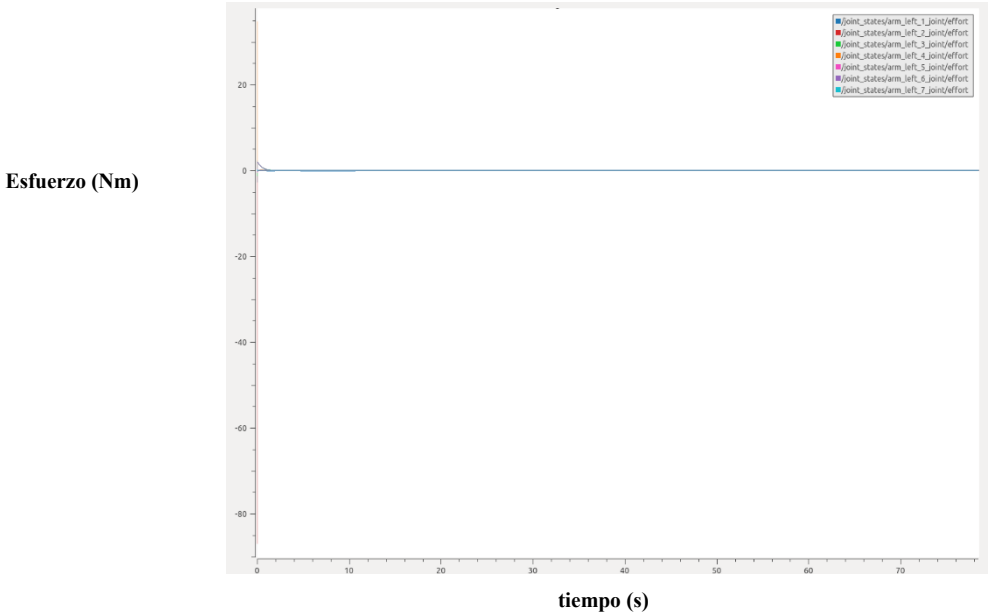



**Figura 5.19:** Esfuerzo realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación de hombro y codo.

En la figura 5.20 puede verse la posición final de las articulaciones del robot en la realización del movimiento. Tal y como puede verse, es necesario mucho más tiempo para conseguir que las articulaciones vayan por el buen camino. Sin embargo, las articulaciones 4, 5 y 6, a pesar de que van por buen camino, todavía les falta un largo camino hasta el objetivo.

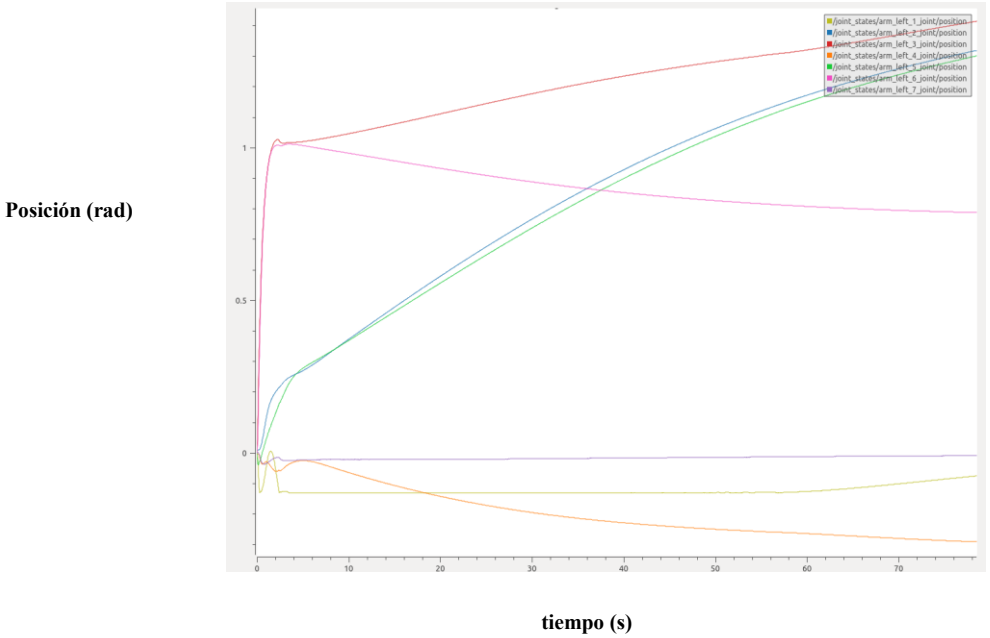



**Figura 5.20:** Posición realizado por el brazo izquierdo con el controlador cartesiano para realizar la rotación de hombro y codo.

En la figura 5.21 puede verse la velocidad de las diferentes articulaciones mientras se realiza la tarea. Como puede verse, además de necesitar más tiempo para estabilizarse en 0, aparecen estas estructuras periódicas nuevamente.

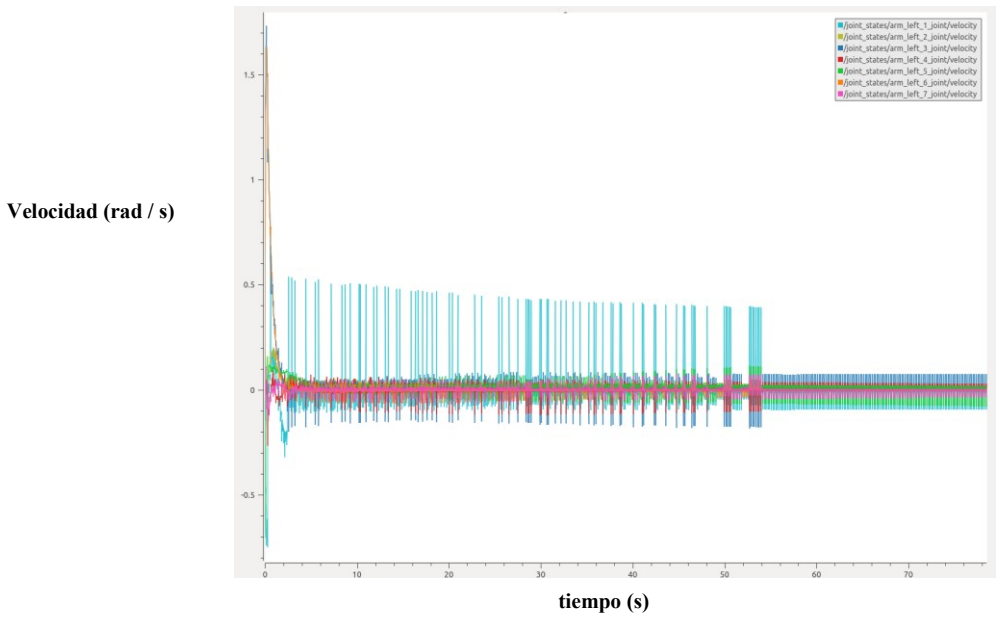



**Figura 5.21:** Velocidades realizadas por el brazo izquierdo con el controlador cartesiano para realizar la rotación de hombro y codo.

# 6. Conclusiones

Este último capítulo de la memoria servirá a modo de conclusión. Tras realizar una introducción al proyecto, se desarrolló el marco teórico de la robótica espacial, introduciendo el robot que serviría de inspiración para desarrollar el proyecto, además del software. A continuación, se vio la metodología seguida para desarrollar el proyecto, para continuar con el desarrollo de los distintos pasos del proyecto. Eso llevaría a presentar los resultados obtenidos para acabar en este apartado de la memoria que pretende ser la conclusión del proyecto mismo.

En la primera parte de la memoria se vieron los objetivos que se plantearon al comienzo de este. Tras ello, se hizo un breve resumen de cómo se encontraba la memoria estructurada.

En la segunda parte del proyecto se definió lo que es un robot, además de los diferentes campos en los que se encuentran presentes. Se profundizó en la robótica espacial de forma breve, ya que el grupo de interés para el proyecto eran los robots humanoides espaciales. Una vez visto cómo van los desarrollos en este campo, se introdujo la empresa española PAL robotics, la cual tiene un desarrollo similar al desarrollado en este proyecto. Tras ello, se comentaron todas las herramientas software empleadas en el proyecto, siendo estas un total de 7 diferentes.

En la tercera parte del proyecto se explicaron los pasos que se siguieron para desarrollar el proyecto. La mayoría de estos pasos deben realizarse en orden ya que, de lo contrario, es complicado realizar el movimiento de un grupo de articulaciones si antes no se ha construido el robot con el que se trabajará.

En la cuarta parte del proyecto se desarrollaron los pasos del apartado anterior, siguiendo estos el desarrollo del entorno de simulación, el diseño del robot y su personalización y su control con varios controladores diferentes, encontrando entre ellos el controlador PD multiarticular con compensación de gravedad o el controlador cartesiano multiarticular.

En la quinta parte del proyecto se presentaron los resultados obtenidos. Para ello, se diseñaron 3 movimientos diferentes, avanzando en dificultad. En la mayoría de ellos, los controladores P fueron los que mejor funcionaron. Si los movimientos abarcaban un mayor rango, era necesario subir la dificultad del control y realizarlo con un controlador PID. En todos ellos se pudo comprobar como en un ambiente de gravedad nula, el controlador PD multiarticular ofrecía el mismo resultado o mejor que el controlador PD multiarticular con compensación de gravedad o el PD multiarticular con compensación precalculada de gravedad.

Para acabar se encuentra esta sexta parte, en la que se analizan las conclusiones que se extraen y los trabajos futuros que se proponen para ser desarrollados. La conclusión que puede extraerse es que, a pesar de desarrollar el proyecto en ROS Kinetic, la mayoría de herramientas necesarias aún se encuentran disponibles. A pesar de ello, los controladores de trayectorias no pudieron desarrollarse al no poder compilar los paquetes necesarios para instalar las herramientas necesarias. No obstante, este no fue el único problema que surgió, aunque el resto de ellos sí pudieron resolverse, no sin antes sudar para lograrlo.

## 6.1. Trabajos futuros

Una vez expuestas las conclusiones, se pueden explorar varios caminos en cuanto a la expansión del proyecto.

Un ejemplo sería subir el proyecto de versión a ROS Melodic. De esta manera podría verse como incorporar los controladores de trayectorias para poder realizar movimientos más complejos, siendo estos más similares a los requeridos en órbita.

Otra ampliación posible sería realizar las pruebas en un entorno real controlado. Algunas universidades cuentan con cámaras que simulan el vacío que sufren los objetos en el espacio. Podrían construirse maquetas de pequeño tamaño para comprobar los resultados extraídos en las simulaciones.

También se podría tratar de que la empresa PAL robotics prestase un robot TALOS y unos manipuladores del robot RREM-C para diseñar el robot real. Sin embargo, esto sería complicado ya que, a la dificultad de la programación, se añadiría la necesidad de realizar las conexiones electrónicas necesarias.

# Bibliografía

[1] Blibley technologies. “How are robots used in space exploration?” 2019. https://blog.bliley.com/robots-used-in-space-exploration

[2] NASA. Robonaut 2. 2019. https://robonaut.jsc.nasa.gov/R2/

[3] W. Khalil and E. Dombre. “Modeling, identification & control of robots”. Hermes Penton Ldt. 2002.

[4] Karel Čapek. “RUR. Robots Universales Rossum: obra en tres actos y un epílogo”. Barcelona: Círculo de lectores. 2004.

[5] Conte Galván R.. “Robots espaciales”. Revista “Hacia el Espacio”, n° 007, 2013. https://haciaelespacio.wixsite.com/haciaelespacio007/robots-espaciales

[6] Raju P., Sikla S., Garg A. and Pandey M.. “A brief review of recent advancement in humanoid robotics research”. Mukt Shabd Journal, Volume IX, Issue VI, 2020.

[7] Howell, E. “Real-life replicants: 6 humanoid robots used for space exploration”. Space.com, 2017. https://www.space.com/38460-humanoid-robots-for-space-exploration.html

[8] Courtney L. “This half-humanoid robot is going to the moon”. Popularmechanics.com, 2020. https://www.popularmechanics.com/technology/robots/a30646158/india-humanoid-robot-vyommitra/

[9] Revista de robots. “Fedor, el soldado robot que regresó del espacio”. Revistaderobots.com, 2020. https://revistaderobots.com/robots-y-robotica/fedor-el-soldado-robot-que-regreso-del-espacio/

[10] Renishaw: apply innovation. “PAL Robotics integra la tecnología de encóderes magnéticos en sus robots para mantener el equilibrio”. https://www.renishaw.es/

[11] PAL Robotics. “Sobre nosotros”. https://pal-robotics.com/

[12] PAL Robotics. “Robots - TALOS”. https://pal-robotics.com/es/robots/talos/

[13] Robots. “TALOS”. https://robots.ieee.org/

[14] Bueckert K. “Robohub opens in University of Waterloo engineering building”. cbc.ca, 2018.

[15] The University of Edinburgh. “TALOS – The humanoid robot”. 2020. https://www.ed.ac.uk/

[16] 20th World Congress of the International Federation of Automatic Control. “Technical visits: LAAS – CNRS – Mo”. 2017. https://www.ifac2017.org/visits.html

[17] ROS. “About ROS”. https://www.ros.org/about-ros/

[18] ROS. “History”. https://www.ros.org/history/

[19] Ragel De La Torre, R. “Modelado, control y simulación de un quadrotor equipado con un brazo manipulador robótico: Capítulo 3: entorno de simulación ROS / Gazebo”. Biblioteca de Ingeniería, Universidad de Sevilla. 2012.

[20] García, G. J. “Apuntes de la asignatura Robótica - materiales extra para prácticas – tema 1”. Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[21] Pomares, J. “Apuntes de la asignatura Sistemas de Control Automático – proyecto”. Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[22] García, G. J. “Apuntes de la asignatura Robótica - materiales extra para prácticas – tema 4”. Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[23] Colomina, O. “Apuntes de la asignatura Robots Móviles – materiales de prácticas – práctica 1”. Grado en ingeniería robótica de la Universidad de Alicante. 2019.

[24] Campusano, M. “ROSBAG”. Laboratorio de robótica y computación del departamento de ciencias computacionales de la Universidad de Chile. 2015.

[25] PlotJuggler. “Página principal”. https://plotjuggler.io/

[26] Tellez, R. “Learn ROS with Python or C++”. Theconstructism.com. 2019. https://www.theconstructsim.com/learn-ros-python-or-cpp/

[27] Williams, D. R. “Planetary Fact Sheet - Metric”. 2019. https://nssdc.gsfc.nasa.gov/

[28] Ingenieria básica. "¿Hay gravedad en la ISS?¿Por qué flotan los astronautas?". 2020. https://ingenieriabasica.es/

[29] Úbeda, A. "Apuntes de la asignatura Control y programación de robots - teoría – robótica humanoide, tema 1". Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[30] Pomares, J. "Apuntes de la asignatura Sistemas de Control Automático – teoría – tema 1". Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[31] García, G. J. "Apuntes de la asignatura Robótica – teoría – tema 4". Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[32] García, G. J. "Apuntes de la asignatura control y programación de robots – teoría – tema 3". Grado en ingeniería robótica de la Universidad de Alicante. 2019.

[33] Jara, C. A. "Apuntes de la asignatura diseño y simulación de robots – teoría – temas 1 al 3". Máster universitario en automática y robótica de la Universidad de Alicante. 2020.

[34] Rust, R., Casas, G., Parascho, S., Jenny, D., Dorfler, K., Helmreich, M., Gandia, A., Ma, Z., Ariza, I., Pacher, M., Lytle, B., Huang, Y. "Robotic fabrication package for the COMPAS Framework". 2018. https://github.com/compas-dev/compas_fab/

# Lista de acrónimos y abreviaturas

| | |
|---|---|
| AILA | Artificial Intelligence Lightweight Android |
| API | Application Programming Interfaces |
| BSON | Binary JSON |
| CBOR | Concise Binary Object Representation |
| COITIA | Colegio Oficial de Ingenieros Técnicos Industriales de Alicante |
| CSV | Comma - Separated Values |
| DARPA | Defense Advanced Research Projects Agency |
| HURO | HUman RObotics |
| I+D | Investigación y Desarrollo |
| IMU | Inertial Measurement Unit |
| ISRO | Indian Space Research Organization |
| ISS | International Space Station |
| JAXA | Japan Aerospace eXploration Agency |
| JSON | JavaScript Object Notation |
| KDL | Kinematics and Dynamics Library |

| | |
|---|---|
| LAAS – CNRS | Laboratoire d'Analyse et d'Architecture des Systèmes - Centre National de la Recherche Scientifique |
| LED | Light - Emitting Diode |
| LIDAR | LIght Detection And Ranging / Laser Imaging Detection And Ranging |
| LTS | Long Term Support |
| MIT | Massachusetts Institute of Technology |
| MQTT | Message Queuing Telemetry Transport |
| NASA | National Aeronautics and Space Administration |
| P | Proportional |
| PD | Proportional and Derivative |
| PID | Proportional, Integral and Derivative |
| PR | Personal Robots |
| RGB | Red Green Blue |
| ROS | Robot Operating System |
| ROSCOSMOS | Russian Federal Space Agency |
| RViz | Ros Visualization |
| STAIR | STanford Artificial Intelligence Robot |
| TFG | Trabajo Fin de Grado |
| TFM | Trabajo Fin de Máster |
| URDF | Unified Robotic Description Format |
| XACRO | Xml mACRO language |
| XML | eXtensible Markup Language |